

AD-A195 315

CDC NOS/BE METED REFERENCE MANUAL(U) DAVID TAYLOR  
RESEARCH CENTER BETHESDA MD COMPUTATION  
MATHEMATICS/LOGISTICS DE PT D V SOMMER 01 JAN 88

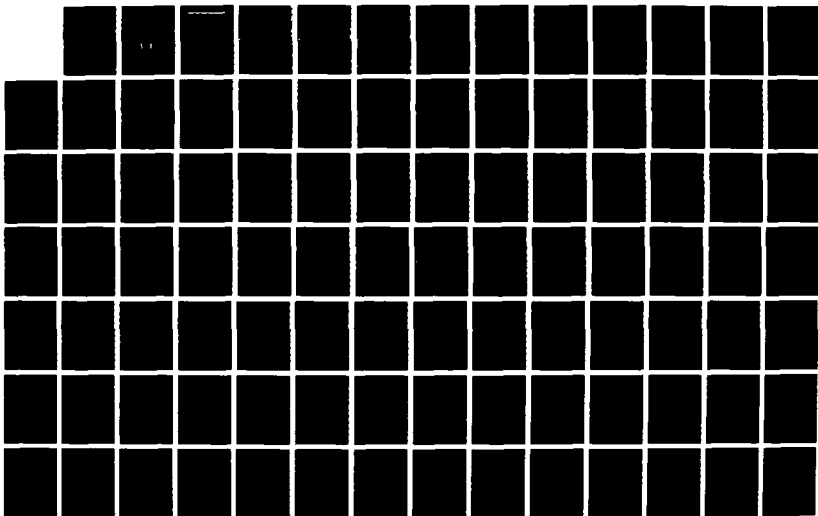
1/2

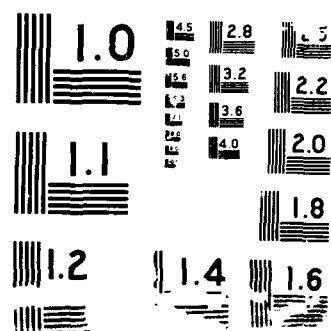
UNCLASSIFIED

DTRC/CHLD-88/2

F/G 12/5

NL





DTIC FILE COPY

(4)

# David W. Taylor Naval Ship Research and Development Center

Bethesda, MD 20084-5000

AD-A195 315

DTRC/CMLD-88/2 January 1988

Computation, Mathematics & Logistics Department  
Departmental Report

CDC NOS/BE NETED REFERENCE MANUAL

by

David V. Sommer

DTIC  
ELECTE  
JUN 07 1988  
S D

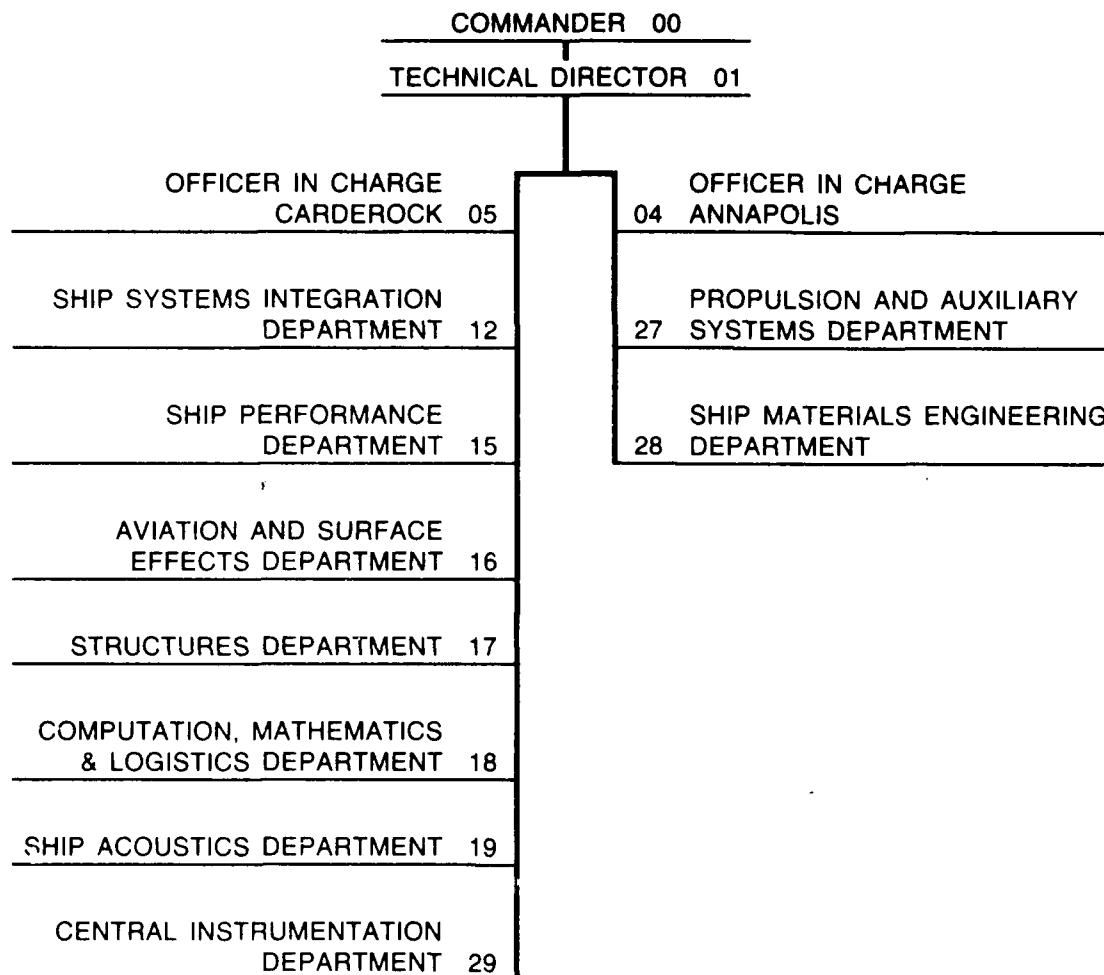
Approved for Public Release; Distribution Unlimited.

DTRC/CMLD-88/2 CDC NOS/BE NETED REFERENCE MANUAL



28 6 7 045

## MAJOR DTNSRDC TECHNICAL COMPONENTS



**DESTRUCTION NOTICE** — For **classified** documents, follow the procedures in DOD 5220.22M, Industrial Security Manual, Section II-9, or DOD 5200.1-R, Information Security Program Regulation, Chapter IX. For **unclassified**, limited documents, destroy by any method that will prevent disclosure of contents or reconstruction of the document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>			1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT <b>Approved for Public Release; Distribution Unlimited</b>		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) <b>CMLD-88/2</b>			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION <b>DTRC</b>		6b. OFFICE SYMBOL (If applicable) <b>1893</b>		7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) <b>Bethesda, MD 20084-5000</b>				7b. ADDRESS (City, State, and ZIP Code)	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)				10. SOURCE OF FUNDING NUMBERS	
PROGRAM ELEMENT NO.		PROJECT NO.		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) <b>CDC NOS/BE NETED Reference Manual</b>					
12. PERSONAL AUTHOR(S) <b>David V. Sommer</b>					
13a. TYPE OF REPORT <b>Final</b>		13b. TIME COVERED FROM <b>010688</b> TO <b>undef</b>		14. DATE OF REPORT (Year, Month, Day) <b>88/01/06</b>	
				15. PAGE COUNT <b>120</b>	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	<b>CDC NOS/BE Editor Text editor</b>		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) <p>The CDC NOS/BE NETED Reference Manual describes the NETED text editor as implemented at DTRC. It is used to prepare computer programs, data files, documents and other publications. Each of the 61 NETED commands is described on a separate page, as well as in the cross-referencing summary by function. A discussion of the history of NETED and its many features, a sample annotated session, and a list of all diagnostic messages are included.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>		
22a. NAME OF RESPONSIBLE INDIVIDUAL <b>David V. Sommer</b>			22b. TELEPHONE (Include Area Code) <b>(301)267-3343</b>		22c. OFFICE SYMBOL <b>1893.1</b>

\*\*\*\*\*

David Taylor Research Center  
Bethesda, Maryland 2084-5000

\*\*\*\*\*  
\*  
\*  
\* Computer Center \*  
\* NETED Reference Manual \*  
\*  
\*  
\*\*\*\*\*

\*\*\*\*\*

by  
David V Sommer

Software Branch  
Code 1893



Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

Computation, Mathematics and Logistics Department  
Departmental Report

January 1988

CMLD-88/2

..

Through Revision 0 (Jan 1988)

..

## \*\*\* Revision Record \*\*\*

RevisionDescription0 (Jan 88)Original printing.

## Contents

Introduction	1-1
NETED at DTRC	1-1
Executing NETED	1-2
All Those NETED Commands	1-3
Files and Operation	1-5
End-of-record and End-of-file	1-5
Output Files	1-5
NETED Modes	1-5
Pointers vs Line Numbers	1-5
The Pointer	1-6
More on the <BOTTOM> and the <TOP>	1-6
Toggle Switches	1-6
Terminal Output -- Optional Responses	1-6
Command Format	1-7
Strings in NETED	1-7
Line Lengths	1-7
Input Mode	1-8
Interrupting NETED	1-8
Special NETED Files	1-8
A Word About Automatic Line Truncation	1-9
Compiling and Running a Program	1-9
 The NETED Commands Alphabetically	 2-1
 A Summary of the NETED Commands by Function	 3-1
Toggle Switches	3-2
Commands Which Move the Pointer	
Without Changing the Edit File	3-3
Print Commands Which Never Move the Pointer	3-4
Commands Which Alter Lines	3-5
File Manipulation Commands	3-7
Commands Which Terminate NETED	3-8
Text/Word Processing Related Commands	3-9
Miscellaneous Commands	3-10

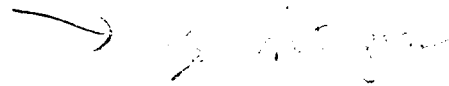
## Appendices

Appendix A: NETED Messages	A-1
Appendix B: Glossary	B-1
Appendix C: Some Hints on Using NETED	C-1
Appendix D: New Features in NETED Version 2.0	D-1



### Abstract

This Manual describes the NETED text editor as implemented on the CDC CYBER Computers at DTRC. NETED is used to prepare computer programs, data files, documents and other publications. Each of the 61 NETED commands is described on a separate page, as well as in the cross-referencing summary by function. A discussion of the history of NETED and its many features, a sample annotated session, and a list of all diagnostic messages are included.



### Administrative Information

The work described in this report was performed in the Software Branch (1893) of the Computation, Mathematics and Logistics Department, David Taylor Research Center, under the sponsorship of the DTRC Computer Center (189).

## \*\*\*\*\* Introduction \*\*\*\*\*

## \*\*\* NETED at DTRC \*\*\*

NETED for the CDC Computers was acquired from Ed Fournier of Lawrence Berkeley Labs, implemented by the Construction Engineering Research Laboratory (CERL), Computer Services Branch, and is maintained by DTRC. CDC NETED is written in Fortran IV Extended, version 4, and Compass (assembly language).

NETED has several advantages over the standard Intercom EDITOR. They include:

- . low cost -- inexpensive to use;
- . does not require line numbering;
- . allows the merging of files;
- . allows much longer search and replacement strings;
- . supports full ASCII upper and lower case files.\*
- . is not a multi-user job; if it does hang, it may be aborted.
- . uses only a small amount of memory (currently under 21K octal).

In 1979, NETED was re-written in Algol for implementation on the Burroughs 7700 computer at DTRC. This version performed as it did on the CDC computers, providing a common editor on Computer Center computers. In addition to the CDC NETED commands, new commands and capabilities were added, increasing the number of commands to 115 and included file backup, system and NETED fault recovery, and many other features.

The last released version of CDC NETED was version 1.4 in March 1981. Work was begun on version 2.0, which was to include a new execute statement and several more commands and features. However, a combination of higher priority work and some implementation problems delayed its release. With the prospect of new computers replacing the CDC CYBERS, version 2.0 never got beyond the test phase.

Since the installation of the replacement computer system has been delayed, the CDC CYBERS will be retained at least a year longer than planned. It was decided to release most of the version 2.0 commands and improvements.

This is expected to be the last released version of NETED.

\* - NETED's ASCII is not compatible with the rest of NOS/BE. After having used NETED for so many years, it would not be practical to change this now.

\*\*\* Executing NETED \*\*\*

To execute NETED, version 2:

```
ATTACH,NETED2
NETED2,<lfn>,<type>,<def_trunc_len>
```

where

<lfn> the local file to be edited. One of:

- . a local file
- . an attached permanent file (may not be overwritten)
- . a non-existent file (to create a new file)

<type> the type of file you are editing; defines the tab columns; sets writes to a permanent file (PF) or queue (Q) device; defines the default truncation length; sets the TRUNC toggle. One of:

value	type	PF/Q	tabs +inc	def_trunc_len
omitted	DATA	PF	7	140
A	ASCII	PF	none	140
C	Cobol	PF	4, 8, 12 +4	72
D	Data	PF	none	140
F	Fortran	PF	7, 10, 13 +3	72
J	Job	PF	none	80
M	coMpass	PF	11, 18, 36 +3	72
P	Pascal	PF	3, 5, 7 +2	72
Q	Q-file	Q	none	140
R	Ratfor	PF	3, 6, 9 +3	72
T	Text	PF	6, 10, 14 +4	80
X	fortran	PF	7, 9, 11 +2	72

<def\_trunc\_len>

the default truncation length; overrides the length implied by <type>; sets the TRUNC toggle. If <def\_trunc\_len> is 0 or invalid, it is set to the maximum line length (140).

Default: that implied by <type>

If <type> and <def\_trunc\_len> are omitted, the default truncation length is set to the maximum line length and the TRUNC toggle is reset.

\*\*\* Leaving NETED \*\*\*

To leave NETED, you must be in Edit Mode. Use the '.' toggle to go from Input Mode to Edit Mode, if necessary. Then type SAVE or QUIT to leave NETED.

## \*\*\* All Those NETED Commands \*\*\*

Don't let the number of NETED commands scare you! After you become familiar with the basic commands, then try a few more. Eventually, you will probably settle on a group which will meet most of your editing needs.

Many commands have similar functions. For instance, consider the search commands. There are two types (the search string must begin in column 1 (F), or in any column (L)). For each of these, the search may be forward or backward (the next letter is B: FB, LB); each of these will search to the next line which meets the test. By adding A to any of the four (FA, FBA, LA, LBA), all lines meeting the test may be listed. Thus, we have 8 search commands: F/FA/FB/FBA, L/LA/LB/LBA.

Similarly, there are four basic ways to write the workfile to disk:

- 1) from top-of-file to the current line (WTOP)
- 2) from the current line for a number of lines (WL)
- 3) the entire workfile (W, SAVE).

The NETED commands are grouped below in 4 levels:

level 0 -- basic commands  
level 1 -- intermediate commands  
level 2 -- advanced commands  
level 3 -- specialized commands

Level 0 (basic):

A/AR	Append (Add on the Right)
B	go to the Bottom of the file
C	Change one string to another
D	Delete lines
F	Find forward (starting in column 1)
FB	Find backward (starting in column 1)
H	list the commands at the terminal
I	Insert a line
L	Locate forward (in any column)
LB	Locate backward (in any column)
N	go N lines forward or backward
P	Print lines
QUIT	QUIT NETED, do not save the edit file
R	Replace a line
RA	print a RANGE of lines around the current line
RC	Repeat the last Change
SAVE	SAVE the edit file and quit NETED
T	go to the Top of the file
W	Write the entire edit file
-	go back one line (same as N -1)
<CR>	(carriage return) go to next line (same as N 1)
.	(period) switch mode (Edit-to-Input or Input-to-Edit)
*	(asterisk) switch "E:"/"I:" prompting on and off

## Level 1 (intermediate)

AL	Add on the Left of the line
DTOP	Delete lines from top-of-file TO (but not including) the current line (the Pointer)
DUP	DUPLICATE the current line
FA	Find forward All occurrences (starting in column 1)
FBA	Find Backward All occurrences (starting in column 1)
LA	Locate forward All occurrences (in any column)
LBA	Locate Backward All occurrences (in any column)
LN	list the Line Number of the current line
M	Merge a file at the pointer
PA	Print All lines
STAB	Set TAB character and columns
UC	Undo the last Change
V	set/reset the Verify toggle
WL	Write Lines to a file
WTOP	Write lines from top-of-file TO the Pointer to a file

## Level 2 (advanced)

J	Join two lines with one space between
JN	Join two lines with No space between
K/KL	Keep the Left-most columns
KR	Keep the Right-most columns
PD	Print lines Double-spaced
PDA	Print All lines Double-spaced
SL	Shift lines to the Left (same as TL)
SR	Shift lines to the Right
TL	Truncate lines on the Left
TR	Truncate lines on the Right

## Level 3 (specialized)

CENTER	CENTER lines
IN	set paragraph INdentation
LEFT	LEFT-justify lines
LEN	list line LENgths
LFN	display/change the Local File Name of the edit file
LM	set Left Margin
LPL	List Page Lengths in the document being edited
PF	future write to *PF device
Q	future write to *Q device
RIGHT	RIGHT-justify lines
RM	set Right Margin
STR	list the current STRing definitions
TI	display the elapsed and remaining TImes
TRUNC	toggle to control TRUNCation
WHAT	list the edit file attributes
X	type a line of column markers (....5...10...)

### \*\*\* Files and Operation \*\*\*

The file to be edited must be a local file; it cannot be connected to your terminal. If it is an attached permanent file, NETED will not allow it to be overwritten. If the file does not exist, NETED will put it onto a permanent file device (\*PF), or a queue device (\*Q) if "NETED,...,Q" is used, when it is written. To change the destination for future writes, use the PF and Q commands.

### \*\* End-of-record and End-of-file \*\*

NETED edits files which contain record marks. When a file is brought under NETED, each record mark is turned into a line consisting of only "EOR" (end-of-record) or "EOF" (end-of-file) as appropriate. In addition, NETED will delete a single trailing EOR, a single trailing EOF, or a trailing EOR/EOF pair.

### \*\* Output Files \*\*

When output, each line consisting solely of "EOR" or "EOF" (in columns 1-3) is converted to its equivalent record mark. An end-of-record is always written at the end.

The file being written is rewound before and after writing, thus replacing any previous contents.

An attached permanent file cannot be overwritten.

### \*\*\* NETED Modes \*\*\*

NETED works in two distinct modes -- Input Mode and Edit Mode. In Input Mode, lines are read from the terminal and entered directly into the edit file. A line consisting of only a period (".") has a special meaning -- switch to Edit Mode.

In Edit Mode, there are a number of commands which allow you to locate lines; add, delete and replace lines; change parts of lines; move your point of interaction with the file ("the Pointer"); insert an entire file at any point in the edit file; save the edit file; and make a copy of all or part of the edit file into another file.

### \*\*\* Pointers vs Line Numbers \*\*\*

NETED is a pointer-oriented editor, unlike the CDC NOS/BE editor, (called EDITOR), which is line-number oriented. With EDITOR, each line must be referred to by its line number. Gaps are left for inserting additional lines. If there is insufficient room, the EDITOR file must be resequenced to make room.

**\*\* The Pointer \*\***

In NETED, your file is conceived of as beginning at the top and extending to the bottom, or end. There is a conceptual pointer, which indicates the current line in the file. If you wish to alter a line, you must first set the pointer on it. When you insert a line, the new line goes after the current line and the new line becomes the current line. The NETED command starts you off with the pointer at the top of your file. When you change between input and Edit Mode -- in either direction -- the pointer does not move. To see which line the pointer is on, just type "P" (in Edit Mode).

The file should not be thought of as circular, in that there is a real difference between having the pointer at the top of the file and having it at the bottom. The top is not the next position after the bottom.

**\*\*\* More on the <BOTTOM> and the <TOP> \*\*\***

In NETED, your file contains two pseudo-lines, the <TOP OF FILE>, and the <BOTTOM OF FILE>. These represent the position of the pointer when it lies just in front of the first line in the file, or just after the last line in the file, respectively. A line inserted with the pointer at the top will go in front of the previous first line -- a line entered at the bottom goes at the end of the file. The pointer then points to the line just entered, and the bottom would be just below it.

Since the top and bottom positions are not really lines, Edit Mode commands which attempt to change them have no effect.

**\*\*\* Toggle Switches \*\*\***

NETED has three commands best thought of as reversing the position of toggle switches. The period is an example. It simply says "change modes". The mode you get depends upon which mode was in effect when the command was issued. Issuing a toggle-type command twice just leaves things as they were. The other toggle-type commands are: V (turns on and off the printing of some commands), and \* (turns on and off the NETED prompt of E: or I:).

**\*\*\* Terminal Output -- Optional Responses \*\*\***

Most of NETED's edit commands produce some kind of printing at your terminal in response. NETED insists upon giving some of these responses, but allows you to disable the printing of others. Those which can be disabled are called "optional responses". The printing or suppressing of optional responses is controlled by the toggle switch command "V". The commands which generate optional responses are the next, locate/find, change, repeat change, undo change commands.

## \*\*\* Command Format \*\*\*

In Edit Mode, NETED will recognize and execute commands from the list in Chapter 2. These commands have one of the following formats:

<command>  
<command> <parameters>

A blank space must separate the <command> from the <parameters> only if the parameter starts with a letter, "?", ".", or "\*". Otherwise, <parameters> may follow the command immediately. <Parameters> must be integers or alphanumeric text. In the following, <n> will indicate an integer parameter, <string> will indicate a string of text. A "\$" is allowed as an integer parameter, and is a shorthand notation for a very large number, which will force most commands to execute to the bottom of the file. Integers which represent a "number of lines" may be preceded by a minus sign ("-") to indicate that after execution of the command, the pointer is to be returned to the line where it started the command.

## \*\*\* Strings in NETED \*\*\*

NETED identifies <string>s in one of two ways. In the "C" command, the <string>s are delimited by a character (except blank) which is not in either <string>. In all other string commands (A/AR, AL, F/L/etc., I, R), the <string> starts after the first blank after the <command> and continues through the last non-blank.

After a <string> has been defined for a particular string command, it may be referenced later as the <string> in any other string command by using an "escape sequence". An escape sequence is the "escape character" (see the ESC command) followed by a single digit. (An escape character is needed because NETED cannot detect the ESC key.) The following are the recognized escape sequences (<e> represents the escape character; default: backslash ("\\")):

<e>0 - the most recent F/FA/FB/FBA/L/LA/LB/LBA <string>  
<e>1 - the most recent C <string1>  
<e>2 - the most recent C <string2>  
<e>3 - the most recent AL <string>  
<e>4 - the most recent A/AR <string>  
<e>5 - the most recent I <string> or the last Input Mode line  
<e>6 - the most recent R <string>  
<e>7 - the second line of the most recent J or JN  
<e>8 - the last line printed by P

## \*\*\* Line Lengths \*\*\*

In NETED, text lines, commands, and patterns may be 140 characters in length.



\*\*\* Input Mode \*\*\*

In NETED's Input Mode, every line entered is taken to be a new line of text to be added to the file at the current position. The pointer is always at the last line of text entered. The one exception is that a line of text which consists of just a period (".") in column 1 is taken as the command to leave Input Mode. If you enter Edit Mode commands while in Input Mode, the commands are accepted as text.

If you enter lines in Edit Mode with the "I" or "R" command, the first two columns are taken up with the command letter and the blank following. Because of the confusion of which column you are in, tabs work only in Input Mode, where the columns you type in correspond to the columns in the text.

NOTE

If a command seems to take forever to execute, it may be that you are really in Input Mode. When NETED enters Input Mode, it displays "INPUT." at your terminal. It indicates Edit Mode by displaying "EDIT.". If in doubt, use the period (".") toggle to see where you were.

\*\*\* Interrupting NETED \*\*\*

Any command which is printing may be interrupted by hitting a key to interrupt it and typing %S to ignore the current print buffer. It may be necessary to repeat this several times if a large number of lines are printing.

If you terminate NETED by %A, your edit file is lost and cannot be recovered.

\*\*\* Special NETED Files \*\*\*

NETED uses the following three files:

\$CONSOL - for displaying at your terminal

\$KEYBRD - for terminal input

NETRNDM - the random edit file

They are returned by NETED at normal termination.

\*\*\* A Word About Automatic Line Truncation \*\*\*

While the longest line possible in NETED is 140 characters, you can have NETED truncate lines earlier. This is especially useful when creating or modifying a program, since compilers ignore all characters after column 72. The "right margin" is used to define the truncation length. Truncation can occur with any command (except CENTER, LEFT, and RIGHT) which changes or inserts lines, including Input Mode; it will NOT occur when lines are read from or written to a file.

If, when you enter NETED, you specify the <type> parameter, the initial and default truncation lengths are defined as one of:

- . 72 (<type>=C, F, M, P, R, X (the languages))
- . 80 (<type>=J or T (job or text and documents))
- . 140 (<type>=A, D, Q, or omitted (ASCII, data and queue files))

If the <def\_trunc\_len> parameter is specified, its value overrides that implied by <type>.

Whether truncation takes place is determined by the setting of the TRUNC toggle. If set, lines are truncated at the right margin; if reset, lines are truncated at the maximum line length (140). When NETED is started, the TRUNC toggle is set if <type> is A, C, D, F, J, M, P, Q, R, T, or X, or if <def\_trunc\_len> is specified. If both <type> and <def\_trunc\_len> are omitted, TRUNC is reset.

To set or reset the TRUNC toggle, use the TRUNC command. To change the truncation length (right margin), use the RM command.

Except for the J and JN commands, truncated data is generally lost and cannot be retrieved.

Both the WHAT command and the NETED prolog will display the default truncation length and whether the TRUNC toggle is set or reset.

\*\*\* Compiling and Running a Program \*\*\*

Since NETED is only a text editor, you must return to NOS/BE command mode to compile and execute a program.

The following sample session illustrates creating a program, compiling and executing it. These symbols are used to annotate the example:

```
s> -- a NOS/BE system response
n> -- a NETED response
o> -- user program output
p> -- a NOS/BE system prompt
u> -- a user entry
<-- -- descriptive comments follow this
```

```

p>  COMMAND-
u>      NETED,myprog,F      <-- create program MYPROG with
                             Fortran tabs and no prompts
n>  NETED 2.0               <-- NETED is running
n>  INPUT.                  <-- new file, Input Mode
u>  C;TEST SQRT
u>      100 CONTINUE
u>      ;PRINT *, 'ENTER A NUMBER, END WITH <PCT>EOF -'
u>      ;READ (5, *, END=200) A
u>      ;PRINT *, 'SQRT(', A, ')=', SQRT(A)
u>      ;GO TO 100
u>      200 CONTINUE
u>      ;END
u>      .                   <-- end of program, go to Edit Mode
n>  EDIT.                   <-- in Edit Mode (no prompting)
u>  PA                      <-- display program just entered
n>  C      TEST SQRT
n>      100 CONTINUE
n>      PRINT *, 'ENTER A NUMBER, END WITH <PCT>EOF -'
n>      READ (5, *, END=200) A
n>      PRINT *, 'SQRT(', A, ')=', SQRT(A)
n>      GO TO 100
n>      200 CONTINUE
n>      END
u>  SAVE                     <-- save the file and leave NETED
n>  MYPROG WRITTEN.
n>  ET=hh:mm:ss CP=hh:mm:ss IO=hh:mm:ss      <-- times used in NETED
p>  COMMAND-
u>      CATALOG,MYPROG,ID=xxxx      <-- to be sure you don't lose it!
p>  COMMAND-
u>      begin,runftn5,,myprog      <-- compile, load and execute;
                                   files INPUT and OUTPUT will
                                   be connected

o>  ENTER A NUMBER, END WITH <PCT>EOF -
u>  25
o>  SQRT(25.0)=5.000000
o>  ENTER A NUMBER, END WITH <PCT>EOF -
u>  50
o>  SQRT(50.0)=7.0710678118
o>  ENTER A NUMBER, END WITH <PCT>EOF -
u>  <--EOF
o>  END MYPROG
p>  COMMAND-

```

## \*\*\*\*\* The NETED Commands Alphabetically \*\*\*\*\*

All NETED commands have the following syntax:

[ <query> ] <command> [ <parameters> ]

where

<query>            A question mark ("?",) immediately preceding <command> causes you to be prompted to verify all changes made by the change commands -or- interrogate some setting in NETED.

<command>        The command to be executed.

<parameters>    The presence of absence of <parameters> depends on the individual command. Generally, if the first character of <parameters> is a non-letter (except "?" and space), then the blank separating it from <command> is not required.

In the examples in the rest of the chapter, each line is preceded by one of:

u>            -- a user entry

-->        -- the position of the pointer after successful completion of the command

<nothing> -- displayed output of the command

In addition,

<--        -- indicates a comment.

A (Append)  
AR (Add on Right)

(Append) A  
(Add on Right) AR

A <string>

AR <string>

Append <string> to the current line (add it to the end of the line).  
If <string> is omitted, the most recent corresponding definition is used.

"<e>3" may be used to refer to the most recent "A/AR" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed line displayed  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: AL, ESC, I, K/KL, KR, P, R, RM, SL, SR,  
STR, TL, TR, TRUNC.

Messages: NO OLD A/AR STRING.  
TRUNCATED.

## Examples:

```

u> P
--> The quick brown fox jumped
u> AR over                                <-- saved as <e>3
--> The quick brown fox jumped over
u> F The game is                          <-- saved as <e>0
--> The game is
u> A                                       <-- uses <e>3
--> The game is over

```

```

u> L rain                                <-- saved as <e>0
--> the rainbow
u> AL <e>3                               <-- saved as <e>4
--> over the rainbow

```

```

u> L under                               <-- saved as <e>0
--> went under the river and
u> C /<e>0/<e>3/                          <-- saved as <e>1
--> went over the river and              and <e>2

```

```

u> STR
CURRENTLY DEFINED STRINGS:
A/AR      005  ESC 3
over
AL        005  ESC 4
over
C 1       006  ESC 1
under
C 2       005  ESC 2
over
F/L       006  ESC 0
under
I         000  ESC 5

J/JN      000  ESC 7

P         026  ESC 8
The quick brown fox jumped
R         000  ESC 6

```

AL (Add on Left)

(Add on Left) AL

AL <string>  
AL

Add <string> at the start of the line. If <string> is omitted, the most recent corresponding definition is used.

"<e>4" may be used to refer to the most recent "AL" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed line displayed  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: A/AR, ESC, I, K/KL, KR, P, R, RM, SL, SR, STR, TL, TR, TRUNC.

Messages: NO OLD AL STRING.  
TRUNCATED.

Examples: See page 2-3.

B (Bottom of file)

(Bottom of file) B

B

Move the pointer to the last line of the file.

Note: NETED at DTRC differs from the specifications for ARPANET editors in the implementation of the B command. At other sites, the B command puts the pointer at the <BOTTOM OF FILE> position and puts you into Input Mode. This was considered dangerous since the last line of the file may be a record mark (EOR or EOF), in which case, lines input after the B command would be separated from the rest of the file by a record mark. If the file being edited were a Fortran program, for example, lines appended after an EOR would not be encountered by the compiler -- a likely source of consternation.

-----

Toggles affecting the output of this statement: V.

V set - bottom-of-file displayed  
V reset - no output

Pointer after command completion: the last line of the file.

Related or similar commands: T.

Messages: <BOTTOM OF FILE> -or- <Bottom of File>  
IMPROPER NETED ARGUMENT.

Examples:

```
u> P -$
--> line 1
    line 2
    ...
    line last
    <BOTTOM OF FILE>
u> B
--> line last
```



C (Change)

(Change) C

```
<?> C <dlm><string1><dlm><string2><dlm> <nlines> <global>
<?> C <dlm><string1><dlm><string2><dlm> <global> <nlines>
```

Change the first or all occurrences of <string1> to <string2> in <nlines> lines.

See also RC (Repeat Change) and UC ("Undo" Change).

<dlm> is a delimiter and may be any character not in either string, except a blank. In most examples in this manual, a slash ("/") is used.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is a dollar sign ("\$"), the change continues to the bottom of the file. If <nlines> is omitted, only the current line is changed. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after any changes, to its position at the start of the change.

<global> is the letter "G". If present, all occurrences of <string1> are changed in each of the <nlines> lines. If omitted, only the first occurrence in each line is changed.

<?> is a question mark ("?"), each change will be displayed and one of the following responses must be given:

- Y - make the change
- C - make the change and continue without question
- N - no change, but proceed with the command
- Q - no change and quit the command now

If <string1> is null (<dlm><dlm>), then <string2> is placed in front of the first position of each of the <nlines> lines. (This is similar to AL <string2>, but allows the same insertion to be made in more than one line and allows trailing blanks to be inserted.)

The end of the line is indicated by ending <string1> with enough blanks to uniquely identify it. For example, to change the final comma to a period in

```
COPYSF,A,OUT,EOI,LR,
```

use

```
C /, /./
```

This will change only the final comma, since the others are not followed by a blank.

If <string2> is null, then <string1> is removed from each of the <nlines> lines.

<nlines> and <global> are optional and may be specified in either order.

Hint: If the options are omitted and the final <dlim> was forgotten, RC will execute the command, eliminating the need to re-type it.

"<e>1" may be used to refer to the most recent "change" <string1> as the <string> in any string command.

"<e>2" may be used to refer to the most recent "change" <string2> as the <string> in any string command.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----  
Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1  
If - <nlines>: unchanged.

Related or similar commands: A/AR, AL, ESC, I, J, JN, K/KL, KR, R, RC, RM, SL, SR, STR, TL, TR, UC.

Messages: \*\*\*GET HELP--COMMAND ABORTED.  
IMPROPER CHANGE ARGUMENT.  
LINE TOO LONG.  
NOTHING CHANGED.  
TRUNCATED.

## Examples:

- 1) u> P  
 --> Corect the line  
 u> C /r/rr/  
 --> Correct the line
  
- 2) u> P -3  
 --> This line has a error.  
 This is ok  
 Here's aother error.  
 u> C / a/ an/ -3  
 --> This line has an error.  
 Here's another error.
  
- 3) u> C 121311G <-- change all 2's to 3's  
 in the current line;  
 the delimiter is "1"
  
- 4) u> C /very long string/new string <-- 3rd delim missing  
 IMPROPER CHANGE ARGUMENT.  
 u> RC <-- Repeat the Change without re-typing
  
- 5) u> C /very long string/new stringG-\$ <-- 3rd delimiter  
 IMPROPER CHANGE ARGUMENT. missing  
 u> C/<e>l/new string/ G -\$ <-- re-enter incorrect part
  
- 6) u> C NO KNOWING <-- change all "O K"s to  
 "OWI" (the delimiter is  
 "N")
  
- 7) u> Suppose you have several lines to enter with only  
 slight variations, such as:  
  
 1 FORMAT (' NOW IS THE TIME FOR ', F5.0, ' TO GO')  
 2 FORMAT (' NOW IS THE TIME FOR ', I5, ' TO GO')  
 ...  
 5 FORMAT (' NOW IS THE TIME FOR ', A, ' TO GO')  
  
 You could just enter, in Input Mode:  
  
 1SF5.0^  
 2SI5^  
 ...  
 5SA^  
  
 and then, with the pointer on the first of line:  
  
 C /S/ FORMAT (' NOW IS THE TIME FOR ', / -5  
 C /^/, ' TO GO')/ 5

CENTER (CENTER within margins)

(CENTER within margins) CENTER

CENTER &lt;nlines&gt;

Center <nlines> lines within the current margins.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is 0 or omitted, only the current line is changed. If <nlines> is a dollar sign ("S"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

If a line is longer than the current margin width, it is not changed.

-----

Toggles affecting the output of this statement: V.

V set	- changed lines are displayed
V reset	- no output

Pointer after command completion:

If <nlines>:	current line + <nlines> - 1.
If - <nlines>:	unchanged.

Related or similar commands: LEFT, LM, RIGHT, RM.

Messages: IMPROPER NETED ARGUMENT.

## Examples:

```
u> P -3
--> First line to be processed
    second line
    line 3
u> ?LM                                <-- display the margins
    LM=001  RM=030  IN=+000  WIDTH=030
u> X 30
--> ....5...10...15...20...25...30
u> RIGHT -3
-->      First line to be processed
                second line
                line 3
u> CENTER -3
-->      First line to be processed
                second line
                line 3
u> LEFT 3
    First line to be processed
    second line
--> line 3
```

D (Delete)

(Delete) D

D &lt;nlines&gt;

Delete <nlines> lines starting with the current line.

If <nlines> is 0, 1, or omitted, only the current line is deleted. If <nlines> is a dollar sign ("\$\$"), the current line and all lines to the bottom of the file are deleted. If <nlines> is negative, no lines are deleted and the current line is displayed.

---

Toggles affecting the output of this statement: none.

Pointer after command completion:

not end of file: the line before the first (or only) line deleted.  
end of file: on the <BOTTOM OF FILE>.

Related or similar commands: DTOP.

Messages: DELETED TO BOTTOM.  
IMPROPER NETED ARGUMENT.

Examples:

```
u> RA 2
-2- line d
-1- line e
--> >>> line f
+1+ line g
+2+ line h
u> D 2
u> RA
-1- line d
--> >>> line e
+1+ line h
```

DTOP (Delete TO Pointer)

(Delete TO Pointer) DTOP

## DTOP

Delete all lines from <top of file> up to, but not including, the current line, which becomes the first line of the file.

---

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: D, WTOP.

Messages: IMPROPER NETED ARGUMENT.

## Examples:

```
u> RA 4
-4- <TOP OF FILE>
-3- line 1
-2- line 2
-1- line 3
--> >>> line 4
-1- line 5
-2- line 6
-3- line 7
-4- line 8
u> DTOP
u> RA 4
-1- <TOP OF FILE>
--> >>> line 4
-1- line 5
-2- line 6
-3- line 7
-4- line 8
```

DUP (DUPLICATE a line)

(DUPLICATE a line) DUP

DUP <ntimes>

Duplicate the current line <ntimes> times following the current line.

If <ntimes> is preceded by a minus sign ("-"), the pointer is not moved.

See page 1-9: A Word About Automatic Line Truncation.

---

Toggles affecting the output of this statement: TRUNC.

TRUNC set - truncation at right margin

TRUNC reset - truncation at maximum line length

Pointer after command completion:

If <ntimes>: last of the duplicated lines.

If - <ntimes>: unchanged (the line being duplicated).

Related or similar commands: D, I, RM.

Messages: CANNOT DUP TOP- OR BOTTOM-OF-FILE.  
IMPROPER NETED ARGUMENT.  
INVALID NUMBER.  
TRUNCATED.



## Examples:

1) u> RA  
-1- line p  
--> >>> line q  
+1+ line r  
u> DUP  
u> RA 2  
-2- line p  
-1- line q  
--> >>> line q  
+1+ line r  
+2+ line s

2) u> RA  
-1- line j  
--> >>> line k  
+1+ line l  
u> DUP -2  
u> RA 3  
-3- line h  
-2- line i  
-1- line j  
--> >>> line k  
+1+ line k  
+2+ line k  
+3+ line l

ESC (define ESCape character)

(define ESCape character) ESC

ESC &lt;e&gt;

?ESC

Define the escape character to be used in <string> substitutions. When NETED is invoked, the escape character is a backslash ("\").

<e> is a single character.

If <e> is omitted, the escape character is deactivated, that is, there is no escape character.

"?ESC" displays the current escape character, if any; <e> is ignored.

See page 1-7: Strings in NETED.

Default at NETED invocation: \ (the backslash)

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: A/AR/AL, F/FB, FA/FBA, I, J, JN, L/LB, LA/LBA, P, R, STR, WHAT.

Messages: ESCAPE CH=<e>  
INVALID ESC ARGUMENT.  
NO ESCAPE CHARACTER.

Examples:

```

1) u> ? ESC          <-- display current escape character
    ESCAPE CH=\
    u> P 2            <-- look at two lines
        The quick brown fox jumped
--> over the lazy dog's back.
    u> C / over the// <-- remove first two words
--> lazy dog's back.
    u> -              <-- back up a line
--> The quick brown fox jumped
    u> STR C1          <-- display first C string
        C 1          ESC 1    009
        over the
    u> A \1            <-- append first C string
--> The quick brown fox jumped over the

```

F (Find)  
FB (Find Backward)

(Find) F  
(Find Backward) FB

F <string>  
FB <string>

Forward or backward search for a <string>. The search begins with the next (or next previous) line.

<string> must start in column 1 of a line. <string>s longer than the maximum line length will, of course, not be found. If <string> is omitted, the most recent "find/locate" <string> is used.

"<e>0" may be used to refer to the most recent "find/locate" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

---

Toggles affecting the output of this statement: V.

V set - each line is listed.  
V reset - no output.

Pointer after command completion:

found : on the line where it found <string>.  
not found: unchanged.

Related or similar commands: ESC, FA/FBA, L/LB, LA/LBA, STR.

Messages: NOT FOUND.

## Examples:

u> RA 4  
-4- Now is the  
-3- time for  
-2- all good men  
-1- to come to the  
--> >>> aid of  
+1+ the quick brown  
+2+ fox who jumped  
+3+ over the lazy  
+4+ dog.

u> F the  
--> the quick brown

u> FB ti  
--> time for

u> F  
NOT FOUND.

<-- still looking for "ti"

FA (Find All)  
FBA (Find Backward All)

(Find All) FA  
(Find Backward All) FBA

FA <string>  
FBA <string>

Forward or backward search for all lines starting with <string>. The search begins with the next (or next previous) line.

<string> must start in column 1 of a line. <string>s longer than the maximum line length will, of course, not be found. If <string> is omitted, the most recent "find/locate" <string> is used.

"<e>0" may be used to refer to the most recent "find/locate" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

---

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: ESC, F/FB, L/LB, LA/LBA, STR.

Messages: NOT FOUND.

## Examples:

```
1) u> RA 4
    -4- Now is
    -3- the time for
    -2- all good men
    -1- to come to the
--> >>> aid of
    +1+ the quick brown
    +2+ fox who jumped
    +3+ over the lazy
    +4+ dog.
    u> FA the
--> the quick brown
    u> FBA
--> the time for
    u> FA time
    NOT FOUND.

2) u> WHAT
    ...
    FORTRAN *PF
    ...
    u> T
--> <TOP OF FILE>
    u> FA C
    (All "C" comments in the Fortran program are
    displayed)
```

HELP (list of NETED commands)

(list of NETED commands) HELP

HELP -or- H

This command displays a list of all NETED commands showing the command syntax and briefly describing its purpose.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: none.

Messages: IMPROPER NETED ARGUMENT.

Examples:

```
u> H
NETED 2.0 COMMANDS
LEAVING NETED:
...
EDITING:
  A STRING      (APPEND)
  AL STRING     (ADD AT LEFT)
...
FILE MANIPULATION:
...
TEXT/WORD PROCESSING RELATED COMMANDS:
...
TOGGLES:
...
INFORMATION AND AIDS:
...
```

I (Insert)

(Insert) I

I &lt;string&gt;

Insert a new line after the current line.

Column 3 of the "I" line is column 1 of the line in the file. Leading blanks (after the blank after the "I") and embedded blanks are significant; trailing blanks are not.

Tabs do not work in the "I" command.

"<e>5" may be used to refer to the most recent "insert" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

---

Toggles affecting the output of this statement: TRUNC.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

Pointer after command completion: at the inserted line.

Related or similar commands: ESC, R, RM, STR.

Messages: TRUNCATED.



## Examples:

```
1)  u> P -3
    --> line q
        line r
        line s
    u> I new line q1
    u> RA
        -1- line q
    --> >>> new line q1
        +1+ line r

2)  Insert a line in several places:
    u> ?ESC
        ESCAPE CH=\
    u> I <line to go many places>
    u> STR I                                <-- display the I string
        I          ESC 5  length
        <line to go many places>
    u> L <next place>                        <-- locate next
    --> <next place>
    u> I \5                                <-- insert without retyping
    u> RA
        -1- <next place>
    --> >>> <line to go many places>
        +1+ <next place + 1>
    u> L <the next place>
    ...
```

IN (set paragraph INdentation)

(set paragraph INdentation) IN

IN &lt;ncols&gt;

"IN" has no effect at  
level 2.0 of NETED.

?IN

Set paragraph indentation relative to LM.

&lt;ncols&gt; may be positive or negative, subject to the restriction that

$$-(LM-1) \leq <ncols> \leq (RM-LM)$$

If <ncols> is \$ or -\$ , the indentation is defined as the largest or smallest value, respectively. That is, (RM-LM) or -(LM-1). If <ncols> is omitted, the paragraph indentation is reset to 0.

Note: <ncols> at this level must be positive or omitted. "IN -2" produces "INVALID NUMBER"; "IN -\$" produces no response.

"?IN" will display the current left and right margins, paragraph indentation, and line width; <ncols> is ignored.

Toggles affecting the output of this statement: V.

V set - LM/RM/IN values are displayed  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: LM, RM.

Messages: INVALID NUMBER.

LM=lll RM=rrr IN=+iii WIDTH=www

Examples:

```

u> ?IN
LM=010 RM=072 IN=+000 WIDTH=063
u> IN 5
LM=010 RM=072 IN=+005 WIDTH=063
u> IN $
LM=010 RM=072 IN=+062 WIDTH=063
u> RM
LM=010 RM=140 IN=+062 WIDTH=131
u> IN $
LM=010 RM=140 IN=+130 WIDTH=131
u> LM
LM=001 RM=140 IN=+130 WIDTH=140

```

J (Join 2 lines - 1 space between) (Join 2 lines - 1 space between) J  
JN (Join 2 lines - no space) (Join 2 lines - no space) JN

J

JN

Join the current line and the following line inserting one space (J) or no space (JN) between them.

Joining will NOT occur if any of the following conditions is present:

- . the pointer is at top-of-file
- . the pointer is at bottom-of-file
- . the pointer is on the last line
- . either of the lines to be joined is an end-of-record (EOR) or an end-of-file (EOF).

"<e>7" may be used to refer to the second of the joined lines as the <string> in any string command.

If the joined lines are truncated, "I <e>7" will put the second line back in (assuming the right margin is large enough).

#### HINT

You may prefer to turn the TRUNC toggle off while joining lines, then use DUP, K, and TL to keep what you want of the first line and delete what you don't want of the second. The WHAT command shows the setting of the TRUNC toggle.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----  
Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: RM.

Messages: CANNOT JOIN EOR/EOF.  
CANNOT JOIN LAST LINE.  
TRUNCATED.

Examples:

- 1) u> P -2  
--> Line 1 to be joined with  
the second line.  
u> J  
--> Line 1 to be joined with the second line.
- 2) u> P -2  
--> Line 1 to be joined with  
the second line.  
u> JN  
--> Line 1 to be joined with the second line.
- 3) u> ?RM  
LM=001 RM=033 IN=+000 WIDTH=033  
u> P -2  
--> Line 1 to be joined with  
the second line.  
u> J  
TRUNCATED.  
--> Line 1 to be joined with the seco  
u> ?ESC  
ESCAPE CH=\  
u> i \7 <-- to restore the truncated second line  
u> RA  
-1- Line 1 to be joined with the seco  
--> >>> the second line.  
+1+ <whatever was there before>

K (Keep left-most columns)  
KL (Keep left-most columns)

(Keep left-most columns) K  
(Keep left-most columns) KL

K <ncols> <nlines>  
KL <ncols> <nlines>

Shorten <nlines> lines by keeping columns on the left and throwing away the rest of each line.

<nlines> is the number of lines to be truncated. It truncates the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is truncated. If <nlines> is a dollar sign ("S"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

#### CAUTION

This is NOT the same as

TR <line\_length-ncols> <nlines>

unless the lines being changed are all the same length.

See page 1-9: A Word About Automatic Line Truncation.

---

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: C, KR, RM, SL, SR, TL, TR.

Messages: INVALID NUMBER.  
SYNTAX ERROR.  
TRUNCATED.

## Examples:

- 1)    u> P  
      --> This line was too long (by ? characters)  
      u> X  
          .....1...10...15...20...25...30...35...40...45...50  
      u> K 22  
      --> This line was too long
- 2)    u> P  
      --> This line was too long (by ? characters)  
      u> X  
          .....1...10...15...20...25...30...35...40...45...50  
      u> KR 22  
      --> long (by ? characters)

KR (Keep right-most columns)

(Keep right-most columns) KR

KR &lt;ncols&gt; &lt;nlines&gt;

Shorten <nlines> lines by keeping columns on the right and throwing away the rest of each line.

<nlines> is the number of lines to be truncated. It truncates the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is truncated. If <nlines> is a dollar sign ("S"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

## CAUTION

This is NOT the same as

TL <line\_length-ncols> <nlines>

unless the lines being changed are all the same length.

See page 1-9: A Word About Automatic Line Truncation.

---

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin

TRUNC reset - truncation at maximum line length

V set - changed lines are displayed

V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.

If - <nlines>: unchanged.

Related or similar commands: C, K/KL, RM, SL, SR, TL, TR.

Messages: INVALID NUMBER.

SYNTAX ERROR.

TRUNCATED.

## Examples:

- 1)    u> P  
      --> This line was too long (by ? characters)  
      u> X  
          ....1...10...15...20...25...30...35...40...45...50  
      u> K 22  
      --> This line was too long
  
- 2)    u> P  
      --> This line was too long (by ? characters)  
      u> X  
          ....1...10...15...20...25...30...35...40...45...50  
      u> KR 22  
      --> long (by ? characters)



L (Locate)  
LB (Locate Backward)

(Locate) L  
(Locate Backward) LB

L <string>  
LB <string>

Forward or backward search for a <string>. The search begins with the next (or next previous) line.

<string> may start in any column of a line. <string>s longer than the maximum line length will, of course, not be found. If <string> is omitted, the most recent "find/locate" <string> is used.

"<e>0" may be used to refer to the most recent "find/locate" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

---

Toggles affecting the output of this statement: V.

V set - each line is displayed  
V reset - no output

Pointer after command completion:

found : on the line where it found <string>.  
not found: unchanged.

Related or similar commands: ESC, F/FB, FA/FBA, LA/LBA, STR.

Messages: NOT FOUND:

Examples:

```
u> RA 4
-4- Now is the
-3- time for
-2- all good men
-1- to come to the
--> >>> aid of
+1+ the quick brown
+2+ fox who jumped
+3+ over the lazy
+4+ dog.
u> L the
--> the quick brown
u> LB
--> to come to the
u> L time
NOT FOUND.
```

LA (Find All)  
LBA (Find Backward All)

(Find All) LA  
(Find Backward All) LBA

LA <string>  
LBA <string>

Forward or backward search for all lines containing <string>. The search begins with the next (or next previous) line.

<string> may start in any column of a line. <string>s longer than the maximum line length will, of course, not be found. If <string> is omitted, the most recent "find/locate" <string> is used.

"<e>0" may be used to refer to the most recent "find/locate" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: ESC, F/FB, FA/FBA, L/LB, STR.

Messages: NOT FOUND.

Examples:

```
1) u> RA 4
    -4- Now is
    -3- the time for
    -2- all good men
    -1- to come to the
    --> >>> aid of
    +1+ the quick brown
    +2+ fox who jumped
    +3+ over the lazy
    +4+ dog.
    u> LA the
        the quick brown
        over the lazy
    u> LBA
        to come to the
        the time for
    u> LA times
        NOT FOUND.
    u> P
    --> aid of
```

<-- pointer is unchanged

2) u> WHAT

...

FORTRAN

...

u> T

--> <TOP OF FILE>

u> FA C

(All "C" comments in the Fortran program are  
displayed)

LEFT (LEFT-justify)

(LEFT-justify) LEFT

LEFT &lt;nlines&gt;

Left-justify <nlines> lines within the current margins.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is 0 or omitted, only the current line is changed. If <nlines> is a dollar sign ("\$\$"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

If a line is longer than the current margin width, it is not changed.

-----

Toggles affecting the output of this statement: V.

V set            - changed lines are displayed  
V reset          - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: CENTER, LM, RIGHT, RM.

Messages: IMPROPER NETED ARGUMENT.

## Examples:

```
u> P -3
--> First line to be processed
      second line
      line 3
u> ?LM                                <-- display the margins
      LM=001 RM=030 IN=+000 WIDTH=030
u> X 30
--> ....5...10...15...20...25...30
u> RIGHT -3
-->      First line to be processed
              second line
              line 3

u> CENTER -3
-->      First line to be processed
              second line
              line 3

u> LEFT 3
      First line to be processed
      second line
--> line 3
```

LEN (list line LENGTH)

(list line LENGTH) LEN

LEN &lt;nlines&gt; &lt;long&gt;

List the length of each of <nlines> lines starting with the current line.

If <nlines> is omitted or invalid, only the length of the current line is listed. If <nlines> is a dollar sign ("\$\$"), the listing continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after listing, to its original position.

If <long> is specified, lines longer than <long> characters will be flagged with an asterisk and a count of the excess characters. If <long> is omitted or is not a number, only the line lengths are listed.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.

If - <nlines>: unchanged.

Related or similar commands: none.

Messages: INVALID NUMBER.

Examples:

```
u> X 30
....5...10...15...20...25...30
u> P
--> this line has 23 chars.
u> LEN
--> 23
u> LEN 1 10      <-- flag if longer than 10 characters
--> 23 * 13
```

LFN (change Local File Name)

(change Local File Name) LFN

LFN &lt;newlfn&gt;

?LFN

Change the name of the edit file to <newlfn>.

If <newlfn> already is a local file, any write/save command to this file will replace its current contents, unless <newlfn> is an attached permanent file, which cannot be overwritten.

"?LFN" will display the current edit file name.

-----

Toggles affecting the output of this statement: V.

V set - the <lfn> is displayed  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: WHAT.

Messages: ILLEGAL FILE NAME.  
LFN: <lfn>

Examples:

```
u> WHAT
  LFN: FILEA
  ...
u> LFN NEWFILE
u> ?LFN
  LFN: NEWFILE
u> SAVE
  NEWFILE WRITTEN.
```

LM (set Left Margin)

(set Left Margin) LM

LM &lt;lm&gt;

?LM

Set the left margin for the CENTER, LEFT and RIGHT commands.

<lm> must be positive and less than the right margin. If <lm> is omitted, the left margin is set to 1.

"?LM" will display the current left and right margins, paragraph indentation, and line width; <lm> is ignored.

-----

Toggles affecting the output of this statement: V.

V set - LM/RM/IN values and the line width are displayed  
V rset - no output

Pointer after command completion: unchanged.

Related or similar commands: IN, NETED, RM.

Messages: INVALID NUMBER.

LM=lll RM=rrr IN=+iii WIDTH=www

Examples:

```
u> ?LM
LM=001 RM=080 IN=+000 WIDTH=080
u> LM 7
LM=007 RM=080 IN=+000 WIDTH=074
u> RM 66
LM=007 RM=066 IN=+000 WIDTH=060
u> LM
LM=001 RM=066 IN=+000 WIDTH=066
u> RM
LM=001 RM=080 IN=+000 WIDTH=080
```



LN (Line Number)

(Line Number) LN

LN

List the line number of the current line.

LN actually counts the lines from the top of the file, so it may use a bit of time when executed far into a long file.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: WHAT.

Messages: ERROR IN LN.  
IMPROPER NETED ARGUMENT.

Examples:

```
u> RA 2
-2- <TOP OF FILE>
-1- line 1
--> >>> line 2
+1+ line 3
+2+ line 4
u> LN
Line      2
```

LPL (List Page Lengths)

(List Page Lengths) LPL

LPL &lt;page\_length&gt;

List page lengths for the document currently in the edit file. The following carriage control (CC) characters in column 1 are recognized:

- 1 - top of page
- blank - print on the next line
- 0 - double space
- - triple space
- + - print on the same line

Separate counts are made of all lines which contain only "EOR" or "EOF" and all other lines (they have bad CCs).

Pages greater than <page\_length> lines are flagged with an asterisk and a count of the number of extra lines on the page.

If <page\_length> is preceded by a minus sign ("-"), bad-CC and EOR/EOF counts for individual pages are not listed. If omitted, 60 is used.

-----

Toggles affecting the output of this statement: V.

- V set - individual page lengths and summary
- V reset - summary only  
(total number of pages, lines, bad CCs, and EOR/EOFs)

Pointer after command completion: unchanged.

Related or similar commands: none.

Messages: <n> BAD CCS.  
<n> EOR/EOFS.  
PG LEN \* >page\_length  
<pg> <len>  
<pg> <len> \* <excess>  
TOTAL: np PGS nl LINES (nt1 TOO LONG)

Examples:

```
u> LPL
PG LEN * >60
1 54
2 27
3 65 * 5
...
15 60
TOTAL: 15 PGS, 724 LINES, (3 TOO LONG)
```

M (Merge file)

(Merge file) M

M &lt;lfm&gt;

Merge all lines from file <lfm> after the current line.

Lines longer than the maximum line length (140) will be truncated. The TRUNC toggle has no effect on reading from a file.

<lfm> must be a local file. It may be an attached permanent file, but may not be a file which is connected to your terminal.

If you get an error message after a merge command and cannot see why the file you specified cannot be merged, it is possible that the file is not local to your session (check the spelling of <lfm>). If you still can't find the problem, you might SAVE the file and use the Intercom FILES command to see what files you have. Note that the merge command will create an empty file, if one doesn't exist, which will show in the output of the FILES command.

Note: There will probably be at least a record mark ("EOR" in your file) at the end of the file you merge. In most cases, you will not want this record mark (or file mark, if present). Therefore, NETED will delete a single trailing EOR, a single trailing EOF, or a trailing EOR/EOF pair. If you really want a record mark there, you must insert it or have extra EORs at the end of the merge file. In any case, you should use the "P" command to be sure the last line merged is what you expect. If you find an EOR or EOF and delete it, keep checking until you are satisfied with the merge.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion:

If any lines merged: at the last (or only) line merged into the edit file.

If no lines merged: unchanged.

Related or similar commands: none.

Messages: <lfm> -- CANNOT READ CONNECTED FILE.

<lfm> EMPTY.

ILLEGAL FILE NAME.

ILLEGAL FILE TYPE.

MERGE REQUIRES A FILE NAME.

<lfm> MERGED.

<lfm> TOO LARGE FOR NETED.

## Examples:

Assume local file M1 contains the following 3 lines:

line M1  
line M2  
line M3

```
u> RA 2
-2- <TOP OF FILE>
-1- line 1
--> >>> line 2
+1+ line 3
+2+ line 4
u> M M1
M1 MERGED.
u> RA 3
-3- line 2
-2- line M1
-1- line M2
--> >>> line M3
+1+ line 3
+2+ line 4
+3+ line 5
```

N (go to Next-th line)  
 <CR> (go to next line)  
 - (go back 1 line)

(go to Next-th line) N  
 (go to next line) <CR>  
 (go back 1 line) -

N <nlines>  
 N -<nlines>  
 <CR>           <-- the RETURN key  
 -

Move the pointer forward or backward <nlines> lines.

If <nlines> is 0, 1, or omitted, the pointer is advanced one line. If <nlines> is a dollar sign ("\$\$"), the pointer is moved to the bottom of the file.

Two short forms are available to move forward (<CR>) or backward (-) one line.

-----  
 Toggles affecting the output of this statement: V.

V set    - the line will be displayed  
 V reset - no output

Pointer after command completion:

N    : current line + <nlines>  
 <CR>: current line + 1  
 -    : current line - 1

Related or similar commands: F/FB, L/LB, P.

Messages: <BOTTOM OF FILE>    or    <Bottom of File>  
           IMPROPER NETED ARGUMENT.  
           <TOP OF FILE>        or    <Top of File>

Examples:

```

u> P -9
--> line 1
    line 2
    ...
    line 9
u> <CR>
--> line 2
u> -
--> line 1
u> N 4
--> line 5
u> N -2
--> line 3

```

P (Print)  
PD (Print Double-spaced)

(Print) P  
(Print Double-spaced) PD

P <nlines>  
PD <nlines>

Print <nlines> lines starting with the current line either single- (P) or double-spaced (PD).

<nlines> is the number of lines to be printed. It prints the current and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is printed. If <nlines> is a dollar sign ("\$"), printing continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after printing, to its original position.

"<e>8" may be used to refer to the last line displayed by the "print" command as the <string> in any string command.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: PA/PDA, RA.

Messages: IMPROPER NETED ARGUMENT.

## Examples:

```
u> P 4
    line n
    line n+1
    line n+2
--> line n+3

u> PD -4
--> line n

    line n+1

    line n+2

    line n+3

u> P $
    <all lines in file from current line on>
--> <BOTTOM OF FILE>

u> P -$
--> line n
    <all remaining lines in file after current line>
    <BOTTOM OF FILE>

u> P
--> line n
```

PA (Print All)  
PDA (Print Double All)

(Print All) PA  
(Print Double All) PDA

PA  
PDA

Print all lines single-spaced (PA) or double-spaced (PDA). <TOP OF FILE> and <BOTTOM OF FILE> are not printed.

---

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: P/PD, RA.

Messages: IMPROPER NETED ARGUMENT.

Examples:

```
u> P
--> This is line number 15
u> PA
    line 1
    line 2
    ...
    last line
u> P
--> This is line number 15
```



PF (write to \*PF)

(write to \*PF) PF

PF

All future write commands (SAVE, W, WL, WTOP) will write to a permanent file device (\*PF - for cataloging). Supercedes the setting implied by the second execute parameter.

-----

Toggles affecting the output of this statement: V

V set - display message  
V reset - do not display message

Pointer after command completion: unchanged.

Related or similar commands: Q, NETED execute statement.

Messages: FUTURE WRITES TO \*PF.

Examples:

```
u> NETED,myprog,Q
--NETED V2.0
MYPROG *Q
INPUT.
...
.
EDIT.
u> WHAT
LFN: MYPROG
MYPROG . *Q
...
u> PF
FUTURE WRITE TO *PF.
u> WHAT
LFN: MYPROG
MYPROG *PF
...
```

Q (write to \*Q)

(write to \*Q) Q

Q

All future write commands (SAVE, W, WL, WTOP) will write to a queue device (\*Q - for routing to an input or output queue). Supercedes the setting implied by the second execute parameter.

-----

Toggles affecting the output of this statement: V

V set - display message  
V reset - do not display message

Pointer after command completion: unchanged.

Related or similar commands: PF, NETED execute statement.

Messages: FUTURE WRITES TO \*Q.

Examples:

```
u> NETED,myjob,F
--NETED V2.0
FORTRAN *PF
INPUT.
...
.
EDIT.
u> WHAT
LFN: MYJOB
FORTRAN *PF
...
u> Q
FUTURE WRITE TO *Q.
u> WHAT
LFN: MYJOB
FORTRAN *Q
...
```

QUIT (leave NETED)

(leave NETED) QUIT

QUIT

Terminate NETED immediately. The edit file is not saved.

If, for example, you wish to alter a given file, but still preserve a copy of the original intact, you could edit it, write the new version using another name, and then quit. Another way is to NETED it, change its name, edit it, and then save it. (See examples 2 and 3.)

---

Toggles affecting the output of this statement: V.

V set - the times are displayed  
V reset - no output

Pointer after command completion: no longer exists.

Related or similar commands: SAVE.

Messages: ET= hh:mm:ss CP= nnn.nnn IO= nnn.nnn  
IMPROPER NETED ARGUMENT.

Examples:

- 1) u> QUIT  
ET= 00:12:52 CP= 001.234 IO= 004.367
- 2) u> NETED,file1  
...  
u> <make changes>  
u> W file2  
FILE2 WRITTEN.  
u> QUIT  
ET= 00:12:52 CP= 001.234 IO= 004.367
- 3) u> NETED,file1  
...  
u> LFN file2  
u> <make changes>  
u> SAVE  
FILE2 WRITTEN.  
ET= 00:12:52 CP= 001.234 IO= 004.367

R (Replace).

(Replace) R

R &lt;string&gt;

Replace the current line.

Column 3 of the "R" line is column 1 of the line in the file. Leading blanks (after the blank after the "R") and embedded blanks are significant; trailing blanks are not.

Tabs do not work in the "R" command.

"R" has the same effect as "D" followed by "I".

"<e>6" may be used to refer to the most recent "replace" <string> as the <string> in any string command.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

Pointer after command completion: unchanged.

Related or similar commands: ESC, I, RM, STR.

Messages: TRUNCATED.

Examples:

```
u> P
--> the current line
u> R A new line.
u> P
--> A new line.
```

RA (RAnge)

(RAnge) RA

RA &lt;nlines&gt;

Display the current line and <nlines> lines on either side. Lines in front are prefixed with "-"; lines after by "+". The current line is prefixed with ">>>".

<nlines> is a single digit. If <nlines> is omitted, one line on either side is displayed. If <nlines> is negative or greater than 9, an error message is displayed.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: P/PD, PA/PDA.

Messages: IMPROPER NETED ARGUMENT.  
INVALID NUMBER.

Examples:

- 1) u> RA  
-1- line n-1  
>>> current line (n)  
+1+ line n+1
- 2) u> RA 4  
-4- line n-4  
-3- line n-3  
-2- line n-2  
-1- line n-1  
>>> current line (n)  
+1+ line n+1  
+2+ line n+2  
+3+ line n+3  
+4+ line n+4

RC (Repeat Change)

(Repeat Change) RC

```
<?> RC <nlines> <global>
<?> RC <global> <nlines>
```

Repeat the last change in a forward direction.

See also C (Change) and UC ("Undo" Change).

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is a dollar sign ("\$"), the change continues to the bottom of the file. If <nlines> is omitted, only the current line is changed. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after any changes, to its position at the start of the change.

<global> is the letter "G". If present, all occurrences of <stringl> are changed in each of the <nlines> lines. If omitted, only the first occurrence in each line is changed.

<?> is a question mark ("?"), each change will be displayed and one of the following responses must be given:

- Y - make the change
- C - make the change and continue without question
- N - no change, but proceed with the command
- Q - no change and quit the command now

<nlines> and <global> are optional and may be specified in either order.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

- TRUNC set - truncation at right margin
- TRUNC reset - truncation at maximum line length
- V set - changed lines are displayed
- V reset - no output

Pointer after command completion:

- If <nlines>: current line + <nlines> - 1
- If - <nlines>: unchanged.

Related or similar commands: A/AR, AL, C, ESC, I, K/KL, KR, R, RM, SL, SR, STR, TL, TR, UC.

Messages: \*\*\*GET HELP--COMMAND ABORTED.  
IMPROPER CHANGE ARGUMENT.  
LINE TOO LONG.  
NOTHING CHANGED.  
TRUNCATED.

Examples:

- 1) u> P  
--> THIS LINE IS TO BE CHANGED.  
u> C /THIS/THAT/  
--> THAT LINE IS TO BE CHANGED.  
u> L THIS  
--> THIS LINE SHOULDN'T BE CHANGED.  
u> L  
--> THIS IS ANOTHER LINE TO BE CHANGED.  
u> RC  
--> THAT IS ANOTHER LINE TO BE CHANGED.
- 2) u> L very long string  
--> THIS LINE HAS A VERY LONG STRING IN IT.  
\* u> C /VERY LONG STRING/NEW STRING <-- 3rd <dlm> omitted  
IMPROPER CHANGE ARGUMENT.  
u> RC  
--> THIS LINE HAS A NEW STRING IN IT.  
u> STR C <-- display the C strings  
C 1 ESC 1 016  
VERY LONG STRING  
C 1 ESC 2 010  
NEW STRING
- \* Since L defines <e>0, this "C" command could have been: "C /\0/NEW STRING", assuming <e> is "\".  
With either form, by the time the end of the line has been scanned, NETED has defined the second change string as "NEW STRING", which is what we wanted.  
Therefore, it is not necessary to retype the change command -- just use RC.

RIGHT (RIGHT-justify)

(RIGHT-justify) RIGHT

RIGHT <nlines>

Right-justify <nlines> lines within the current margins.

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is 0 or omitted, only the current line is changed. If <nlines> is a dollar sign ("\$"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

If a line is longer than the current margin width, it is not changed.

-----

Toggles affecting the output of this statement: V.

V set           - changed lines are displayed  
V reset         - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: CENTER, LEFT, LM, RM.

Messages: IMPROPER NETED ARGUMENT.



## Examples:

```
u> P -3
--> First line to be processed
      second line
      line 3
u> ?LM                                <-- display the margins
      LM=001 RM=030 IN=+000 WIDTH=030
u> X 30
--> .....5...10...15...20...25...30
u> RIGHT -3
-->      First line to be processed
                        second line
                        line 3
u> CENTER -3
-->      First line to be processed
                        second line
                        line 3
u> LEFT 3
      First line to be processed
      second line
--> line 3
```

RM (set Right Margin)

(set Right Margin) RM

RM &lt;rm&gt;

?RM

Set the right margin for CENTER, LEFT and RIGHT commands. <rm> is also the truncation length when the TRUNC toggle is set.

<rm> must be positive, greater than the left margin, and less than the maximum line length. If <rm> is omitted, the right margin is set to the maximum line length (currently 140).

"?RM" will display the current left and right margins, paragraph indentation, and line width; <rm> is ignored.

See page 1-9: A Word About Automatic Line Truncation.

---

Toggles affecting the output of this statement: V.

V set - LM/RM/IN values and the line width are displayed  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: IN, LM, NETED, TRUNC.

Messages: INVALID NUMBER.

LM=lll RM=rrr IN=+iii WIDTH=www

Examples:

```
u> ?LM
LM=001 RM=080 IN=+000 WIDTH=080
u> LM 7
LM=007 RM=080 IN=+000 WIDTH=074
u> RM 66
LM=007 RM=066 IN=+000 WIDTH=060
u> LM
LM=001 RM=066 IN=+000 WIDTH=066
u> RM
LM=001 RM=080 IN=+000 WIDTH=080
```

SAVE (SAVE edit file, leave NETED) (SAVE edit file, leave NETED) SAVE

SAVE <lf>  
SAVE

The edit file is written to local file <lf> and NETED is terminated.

If <lf> is omitted, the <lf> used in the NETED execute statement or the most recent LFN command is used.

If <lf> is an attached permanent file, the SAVE is not done and NETED continues.

Note: SAVE is the same as W followed by QUIT.

#### CAUTION

At this level of NETED, you can still SAVE to a connected file.

-----

Toggles affecting the output of this statement: V.

V set - the execution times are displayed  
V reset - times are not displayed

Pointer after command completion: no longer exists.

Related or similar commands: LFN, QUIT, W.

Messages: <lf> - CANNOT OVERWRITE PERM FILE.  
EDIT FILE EMPTY.  
ET= hh:mm:ss CP= nnn.nnn IO= nnn.nnn  
ILLEGAL FILE NAME.  
ILLEGAL FILE TYPE.  
<lf> WRITTEN.

#### Examples:

```
u> NETED,MYPROG
...
u> SAVE
MYPROG WRITTEN.
ET=<et> CP=<cp> IO=<io>

u> V
u> SAVE ABC
ABC WRITTEN.
```

SL (Shift Left)

(Shift Left) SL

SL &lt;ncols&gt; &lt;nlines&gt;

Shift lines to the left by deleting <ncols> characters from the start of each line.

<nlines> is the number of lines to be shifted. It shifts the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is shifted. If <nlines> is a dollar sign ("\$"), shifting continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

SL is a synonym for TL.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: C, K/KL, KR, LM/LEFT, RM, SR, TL, TR.

Messages: INVALID NUMBER.  
SYNTAX ERROR.  
TRUNCATED.

Examples:

```
u> P
--> Line to be shifted.
u> SR 5
--> Line to be shifted.
u> SL 10
--> to be shifted.
```

SR (Shift Right)

(Shift Right) SR

SR &lt;ncols&gt; &lt;nlines&gt;

Shift lines to the right by adding <ncols> blanks to the start of each line.

<nlines> is the number of lines to be shifted. It shifts the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is shifted. If <nlines> is a dollar sign ("\$"), shifting continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

This is shorthand for "C //<ncols\_blanks>/ <nlines>".

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: C, K/KL, KR, LM/LEFT, RM, SL, TL, TR.

Messages: INVALID NUMBER.  
SYNTAX ERROR.  
TRUNCATED.

Examples:

```
u> P
--> Line to be shifted.
u> SR 5
-->      Line to be shifted.
u> SL 10
--> to be shifted.
```

(Set TAB) STAB

STAB <tabch>	<tab1>,<tab2>,<tab3>,...	<-- set tab and stops
STAB <tabch>	<tab1>,<tab2>...+<inc>	<-- set tab and stops and tab increment
STAB <tabch>		<-- set tab
STAB <type>		<-- set pre-defined stops
STAB		<-- reset stops
?STAB		<-- display <tabch>

Define the tab character, <tabch>, and up to 30 tab columns, either specifically, or as at least one tab column and an increment.

<type> has the same meaning as in the NETED execute statement. The following upper case letters are recognized as pre-defined tabs:

A (ASCII)	- no tabs (will NOT change the edit file)
C (Cobol)	- 8+4 - 8,12,16, 20, 24, 28, 32, 36, 40, 44, 48,52,56, 60, 64, 68, 72, 76, 80, 84, 88,92,96,100,104,108,112,116,120,124
D (Data)	- no tabs
F (Fortran)	- 7+3 - 7,10,13,16,19,22,25,28,31,34, 37,40,43,46,49,52,55,58,61,64, 67,70,73,76,79,82,85,88,91,94
J (Job)	- no tabs
M (coMpass)	- 11,18,36+3 - 11,18,21,24,27,30,33,36,39,42, 45,48,51,54,57,60,63,66,69,72, 75,78,81,84,87,90,93,96,99,102
P (Pascal)	- 3+2 - 3, 5, 7, 9,11,13,15,17,19,21, 23,25,27,29,31,33,35,37,39,41, 43,45,47,49,51,53,55,57,59,61
Q (Job)	- no tabs
R (Ratfor)	- 3+3 - 3, 6, 9,12,15,18,21,24,27,30, 33,36,39,42,45,48,51,54,57,60, 63,66,69,72,75,78,81,84,87,90
T (Text)	- no tabs
X (Fortran)	- 7+2 - 7, 9,11,13,15,17,19,21,23,35, 37,39,41,43,45,47,49,51,53,55, 57,59,61,63,65,67,69,71,73,75

<tabch> may be any character except a blank (" "), RETURN (<CR>), ctrl-X, ctrl-U, ctrl-H, DEL, and the upper case letters A, C, D, F, J, M, P, Q, R, T, and X. Full ASCII characters may be used to indicate a <tabch>, even if the file is in Display Code.

In the tab list, <tabl> may not be 1. If there are any tabs, there must be at least 1 ("<inc>" by itself is not allowed) and there will always be 30 of them. If fewer than 30 tabs are specified, additional tabs are set every 3 positions following <tabn>. If the tab list is invalid (non-digit, non-ascending numbers, a tab after an increment, etc.), the tabs are not changed. If the tab list is omitted, the most recent tab settings will be used.

<inc> is the increment to use after the last specified tab column to fill out the 30 tabs. For example, "STAB ; 4,7,10+5" sets tabs at columns 4, 7, 10, 15, 20, 25, ..., 140.

If <tabch> and the tab list are omitted, <tabch> will not be changed. The tab settings will be set to those for the <type> you specified in the NETED execute statement.

"?STAB" will display the current tab character.

See page 1-2 (second execute parameter).

-----

Toggles affecting the output of this statement: V.

V set - actual number of tabs if < 30  
V reset - no output

Pointer after command completion: unchanged.

Related or similar commands: WHAT.

Messages: BAD TAB LIST.  
ERROR IN STAB.  
<n> TABS.  
TAB CH=<tabch>

## Examples:

```
u> NETED,<lfm>,C      <-- set Cobol tabs (8+4)

u> ?STAB
  TAB CH=;

u> STAB F              <-- set Fortran tabs (7+3)

u> ?STAB
  TAB CH=;

u> STAB ; 6,11,16,21   <-- tab is ";", tabs are at 6, 11,
                        16, 21, 24, 27, 30, ..., 99

u> ?STAB
  TAB CH=;

u> STAB ^              <-- change <tabch>

u> ?STAB
  TAB CH=^

u> STAB                <-- reset the tabs to what was
                        implied by the second NETED
                        execute parameter (in this
                        example)

u> ?STAB
  TAB CH=^

u> STAB ~              <-- assuming the above sequence,
                        the tab character is now "~"
                        with tabs at 6, 11, 16, 21,
                        24, 27, 30, ..., 99

u> ?STAB
  TAB CH=~

u> STAB ~ 6+5          <-- tabs set at 6, 11, 16, 21,
  27 TABS.             ..., 136

u> STAB ; 11+10        <-- the tab character is ";" with
  13 TABS.             tabs set at 11, 21, ..., 131
```



STR (list current STRings)

(list current STRings) STR

STR

STR &lt;cmd&gt;

List the current definitions of the strings which can be remembered by NETED: A/AR, AL, C (2 strings), F/FA/FB/FBA/L/LA/LB/LBA (1 string for all), I, R, P (the last line printed).

The list, which includes the "escape" sequence which may be used to refer to each string, has the following format:

## CURRENTLY-DEFINED STRINGS:

```

F/L/...    <length> ESC 0
<the find/locate string>
C 1        <length> ESC 1
<the change string1>
C 2        <length> ESC 2
<the change string2>
A/AR       <length> ESC 3
<the append string>
AL         <length> ESC 4
<the add_left string>
I          <length> ESC 5
<the insertstring>
J/JN       <length> ESC 7
<the second joined line>
P          <length> ESC 8
<the last printed line>
R          <length> ESC 6
<the replace string>

```

If <cmd> is specified, only that definition is displayed. <cmd> may be one of: A, AL, AR, C1, C2, C (both C strings), F, FA, FB, FBA, I, J, JN, L, LA, LB, LBA, P, or R.

See page 1-7: Strings in NETED.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: A/AL/AR, C, ESC, F/FB, FA/FBA, I, J/JN, L/LB, LA/LBA, P, R, WHAT.

Messages: INVALID STR REQUEST.

## Examples:

u&gt; STR

## CURRENTLY-DEFINED STRINGS:

F/L/... 22 ESC 0

the last search string

C 1 9 ESC 1

abcdefghi

C 2 0 ESC 2

A/AR 19 ESC 3

the &lt;string test&gt;.

AL 2 ESC 4

of

I 25 ESC 5

This is an inserted line.

J/JN 29 ESC 7

line joined to preceding line

P 22 ESC 8

The last printed line.

R 19 ESC 6

A replacement line.

u&gt; STR C

C 1 9 ESC 1

abcdefghi

C 2 0 ESC 2

T (Top of file)

(Top of file) T

T

Move the pointer to the top of the file.

-----

Toggles affecting the output of this statement: V.

V set - top-of-file displayed  
V reset - no output

Pointer after command completion: the last line of the file.

Related or similar commands: B.

Messages: IMPROPER NETED ARGUMENT.

<TOP OF FILE> -or- <Top of File>

Examples:

```
u> P -$
--> line 19
    line 20
    ...
    line last
    <BOTTOM OF FILE>
u> T
--> <TOP OF FILE>
```

TI (Times)

(Times) TI

TI

Display the elapsed, processor and I/O times used and the processor time remaining:

ET= hh:mm:ss CP= nnn.nnn IO= nnn.nnn  
REMAINING CP= nnn.nnn

If you have a lot of editing to do, "W" the file frequently and use "TI" to watch your remaining CP time. If time runs out, NETED will abort and you will lose all the editing you have done since the last time you wrote the file. When you get down to just a few seconds, SAVE the file (and leave NETED) and then go back into NETED to continue editing with additional time.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: none.

Messages: ET= hh:mm:ss CP= nnn.nnn IO= nnn.nnn  
IMPROPER NETED ARGUMENT.  
REMAINING CP= nnn.nnn

Examples:

u> TI  
ET= 00:12:52 CP= 001.234 IO= 004.367  
REMAINING CP= 018.766

TL (Truncate on the Left)

(Truncate on the Left) TL

TL &lt;ncols&gt; &lt;nlines&gt;

Shorten lines by truncating columns on the left.

<nlines> is the number of lines to be truncated. It truncates the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is truncated. If <nlines> is a dollar sign ("S"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

TL is a synonym for SL.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: C, K/KL, KR, RM, SL, SR, TR.

Messages: INVALID NUMBER.  
SYNTAX ERROR.  
TRUNCATED.

Examples:

```
u> P
--> This line was too long (by 18 characters)
u> TR 18
--> This line was too long
```

```
u> P
--> This line was too long (by 18 characters)
u> TL 18
--> long (by 18 characters)
```

TR (Truncate on the Right)

(Truncate on the Right) TR

TR &lt;ncols&gt; &lt;nlines&gt;

Shorten lines by truncating columns on the right.

<nlines> is the number of lines to be truncated. It truncates the current line and the next <nlines>-1 lines. If <nlines> is omitted, only the current line is truncated. If <nlines> is a dollar sign ("\$\$"), the change continues to the bottom of the file. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after the change, to its original position.

See page 1-9: A Word About Automatic Line Truncation.

-----

Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output

Pointer after command completion:

If <nlines>: current line + <nlines> - 1.  
If - <nlines>: unchanged.

Related or similar commands: C, K/KL, KR, RM, SL, SR, TL.

Messages: INVALID NUMBER.  
SYNTAX ERROR.  
TRUNCATED.

Examples:

```
u> P
--> This line was too long (by 18 characters)
u> TR 18
--> This line was too long

u> P
--> This line was too long (by 18 characters)
u> TL 18
--> long (by 18 characters)
```

.TRUNC (change truncation toggle) (change truncation toggle) TRUNC

## TRUNC

Turn automatic line truncation on or off.

When the TRUNC toggle is set, input lines are truncated at the right margin. When reset, lines are truncated at the maximum line length.

The WHAT command shows the setting of this toggle.

See page 1-9: A Word About Automatic Line Truncation.

-----

Default: If <type> or <def\_trunc\_len>  
          is specified in NETED statement - TRUNC is set  
          If both are omitted - TRUNC is reset

Toggles affecting the output of this statement: none.

Commands this toggle affects: A/AR, AL, C, DUP, I, J, JN, K/KL, KR, R,  
                                  RC, SL, SR, TL, TR, UC, and Input Mode.

Pointer after command completion: unchanged.

Related or similar commands: NETED, RM, WHAT.

Messages: none.

### Examples:

```
1)  u> NETED,NEWFILE,D,10
      DATA      *PF
      DEFAULT TRUNCATION LENGTH=010
      ...
      INPUT.
      u> line 1
      u> line 2
      u> a too-long line
      TRUNCATED.
      u> .
      EDIT.
      u> P
      --> a too-long
```

UC ("Undo" Change)

("Undo" Change) UC

<?> UC <nlines> <global>  
<?> UC <global> <nlines>

"Undo" the last change. It is shorthand for  
C /<e>2/<e>1/  
but does not change the values of <e>1 or <e>2.

See also C (Change) and RC (Repeat Change).

<nlines> is the number of lines to be changed. It changes the current line and the next <nlines>-1 lines. If <nlines> is a dollar sign ("S"), the change continues to the bottom of the file. If <nlines> is omitted, only the current line is changed. If <nlines> is preceded by a minus sign ("-"), the pointer is returned, after any changes, to its position at the start of the change.

<global> is the letter "G". If present, all occurrences of <string1> are changed in each of the <nlines> lines. If omitted, only the first occurrence in each line is changed.

<?> is a question mark ("?"), each change will be displayed and one of the following responses must be given:

- Y - make the change
- C - make the change and continue without question
- N - no change, but proceed with the command
- Q - no change and quit the command now

#### CAUTION

Use with care because "<e>2" may appear in more than just the places you changed and may cause the wrong string(s) to be changed.

See page 1-7: Strings in NETED.

See page 1-9: A Word About Automatic Line Truncation.

-----  
Toggles affecting the output of this statement: TRUNC, V.

TRUNC set - truncation at right margin  
TRUNC reset - truncation at maximum line length

V set - changed lines are displayed  
V reset - no output



## Pointer after command completion:

If <nlines>: current line + <nlines> - 1  
 If - <nlines>: unchanged.

Related or similar commands: A/AR, AL, C, ESC, I, J, JN, K/KL, KR, R,  
 RC, RM, SL, SR, STR, TL, TR.

Messages: \*\*\*GET HELP--COMMAND ABORTED.  
 IMPROPER CHANGE ARGUMENT.  
 LINE TOO LONG.  
 NOTHING CHANGED.  
 TRUNCATED.

## Examples:

```

1)  u> P
    --> This line hss an error.
    u> C/s/a/                                <-- changes the wrong "s"
    --> Thia line hss an error.
    u> UC                                    <-- restore the line
    --> This line hss an error.
    u> C/ss/as/                             <-- the correct change
    --> This line has an error.

1)  u> P
    --> The first shall be last
    u> C /la/fir/
    --> the first shall be first
    u> UC
    --> the last shall be first

2)  u> P -3
    --> This line has a error.
        This is ok
        Here's aother error.
wrong: u> C /a/s/ -3                        <-- we're making an incorrect change
    --> This line hss a error.
        This is ok
        Here's sother error.
    u> UC -3                                <-- we want to undo the last change
    --> Thia line hss a error.              <-- notice that the wrong
        Thia is ok                          <-- changes were made
        Here'a sother error.

right: u> C/ a/ an/ -3                      <-- we should have used
    --> This line has an error.            <-- a specific change
        This is ok
        Here's another error.
  
```

V (Verify toggle)

(Verify toggle) V

V

Flip the verify toggle switch.

When the verify toggle is set, all generated NETED lines are displayed.

When the verify toggle is reset, lines relating to changes, pointer movement and file reading and writing are not displayed.

Error messages will always be displayed.

The WHAT command shows the setting of this toggle.

-----

Default: set.

Toggles affecting the output of this statement: none.

Commands this toggle affects: A/AR, AL, B, C, CENTER, F/FB, FA/FBA, IN,  
K/KL, KR, L/LB, LA/LBA, LEFT, LFN, LM,  
LPL, N, RIGHT, RM, SL, SR, STAB, TL, TR.

Pointer after command completion: unchanged.

Related or similar commands: WHAT.

Messages: none.

## Examples:

```
u> P
--> line 1

u> WHAT
...
TOGGLES SET: V * TRUNC
...
u> V                <-- to reset it
u> N+5
u> P

u> WHAT
...
TOGGLES SET:      TRUNC
u> V                <-- to set it
u> N+5
--> line 11
```

W (Write)

(Write) W

W &lt;lfm&gt;

The entire edit file is written into local file <lfm>.

If <lfm> is omitted, the <lfm> from the most recent LFN command or, if none, the NETED execute statement is used.

If <lfm> is an attached permanent file, an error message is displayed and the write will not take place.

Suggestion: Write your edit file every now and then. That way, if you make a disastrous change, you will have a fairly recent version to return to. There scratch files are not, however, protected from system failures.

#### CAUTION

At this level of NETED, you can still write to a connected file.

-----  
Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: LFN, SAVE, WL, WTOP.

Messages: <lfm> - CANNOT OVERWRITE PERM FILE.  
EDIT FILE EMPTY.  
ILLEGAL FILE NAME.  
ILLEGAL FILE TYPE.  
<lfm> WRITTEN.

#### Examples:

```
u> NETED,MYFILE
u> ...
u> W
    MYFILE WRITTEN.

u> ATTACH,MYPROG,ID=xxxx.
u> NETED,MYPROG,F
u> ...
u> W
    CANNOT OVERWRITE PERM FILE - MYPROG
u> W NEWFILE
    NEWFILE WRITTEN.
```

WHAT (edit file attributes)

(edit file attributes) WHAT

WHAT

List information about the edit file and the NETED session.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: none.

Messages: IMPROPER NETED ARGUMENT.

Examples:

```
u> WHAT
  LFN:*<lfm>
  <type> *PF          <-- e.g., FORTRAN
  DEFAULT TRUNCATION LENGTH=140
  473 LINES
  TOGGLES SET: V * TRUNC
  LM=001 RM=072 IN=+000 WIDTH=072
  ESCAPE CH=\          -or- NO ESCAPE CHARACTER.
  TAB CH=;             -or- NO TAB CHARACTER.
```

WL (Write Lines)

(Write Lines) WL

WL &lt;nlines&gt; &lt;lfn&gt;

Write <nlines> lines, starting with the current line, to local file <lfn>.

If <nlines> is omitted, one line is written. If <nlines> is a dollar sign ("S"), all lines from the current line to the bottom of the file are written.

<lfn> is required. If <lfn> is an attached permanent file, the write will not take place.

WL is useful in copying or moving lines in the edit file.

## CAUTION

At this level of NETED, you can still write to a connected file.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: SAVE, W, WTOP.

Messages: <lfn> - CANNOT OVERWRITE PERM FILE.  
ILLEGAL FILE NAME.  
ILLEGAL FILE TYPE.  
NO LINES TO WRITE.  
WL REQUIRES A FILE NAME.  
<lfn> WRITTEN.

## Examples:

- 1) u> WL 26 FILE1  
FILE1 WRITTEN.
- 2) Copy 5 lines at another place:  
u> L <first line to be copied>  
--> <first line to be copied>  
u> WL 5 TEMP  
TEMP WRITTEN.  
u> L <location after which lines are to be inserted>  
--> <location after which lines are to be inserted>  
u> M TEMP  
TEMP MERGED.
- 3) Move 5 lines to another place:  
u> L <first line to be copied>  
--> <first line to be copied>  
u> WL 5 TEMP  
TEMP WRITTEN.  
u> D 5 <-- be sure to delete them  
u> L <location after which lines are to be inserted>  
--> <location after which lines are to be inserted>  
u> M TEMP  
TEMP MERGED.

WTOP (Write TO Pointer)

(Write TO Pointer) WTOP

WTOP &lt;lfn&gt;

Write all lines from the top of the file up to, but not including, the current line into local file <lfn>.

<lfn> is required. If <lfn> is an attached permanent file, the write will not take place.

## CAUTION

At this level of NETED, you can still write to a connected file.

-----

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: DTOP, SAVE, W, WL.

Messages: <lfn> - CANNOT OVERWRITE PERM FILE.  
EDIT FILE EMPTY.  
ILLEGAL FILE NAME.  
ILLEGAL FILE TYPE.  
NO TEXT BEFORE CURRENT LINE.  
<lfn> WRITTEN.  
WTOP REQUIRES A FILE NAME.

## Examples:

u> WTOP FILE1  
FILE1 WRITTEN.



X (column markers)

(column markers) X

X <ncols>

A line is displayed marking <ncols> columns.

If <ncols> is omitted, the value used is determined by the TRUNC toggle (see below).

-----

Toggles affecting the output of this statement: TRUNC.

TRUNC set - if <ncols> is omitted, the right margin is used

TRUNC reset - if <ncols> is omitted, the default truncation length (<def\_trunc\_len> or that implied by <type> in the NETED statement is used

Pointer after command completion: unchanged.

Related or similar commands: none.

Messages: INVALID NUMBER.

Examples:

u> X 35  
....5...10...15...20...25...30...35

u> X  
....5...10...15...20...25...30...35 ... 65...70..

. (change mode toggle)

(change mode toggle) .

Change the mode of NETED from Edit-to-Input or from Input-to-edit. The name of the mode being entered is always displayed in response to this command.

When in Edit Mode, all NETED commands may be entered.

When in Input Mode, this command is the only NETED command recognized.

To enter a line consisting of just a period, use "I" or "R". In Input Mode, enter a line with something else and change it later.

-----  
Default: If the file being edited exists: Edit Mode.  
          If the file does not exist : Input Mode.

Toggles affecting the output of this statement: none.

Pointer after command completion:  
  Edit-to-Input: unchanged.  
  Input-to-Edit: at the last line entered in Input Mode.

Related or similar commands: none.

Messages: EDIT.       -or-     Edit.  
          INPUT.      -or-     Input.

AD-A195 315

CDC NOS/BE NETED REFERENCE MANUAL(U) DAVID TAYLOR  
RESEARCH CENTER BETHESDA MD COMPUTATION  
MATHEMATICS/LOGISTICS DE PT D V SOMMER 01 JAN 88  
DTRC/CHLD-88/2

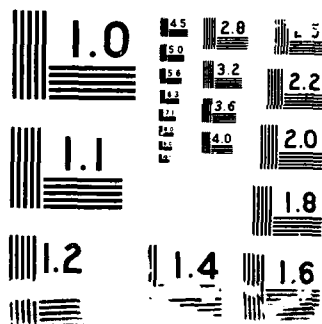
272

UNCLASSIFIED

F/G 12/5

NL





## Examples:

- 1) u> NETED,NEWFILE  
...  
INPUT.  
u> line 1  
u> line 2  
u> line 3  
u> .  
EDIT.  
u> P  
--> line 3  
u> .  
INPUT.
- 2) u> NETED,OLDFILE  
...  
EDIT.  
u> PA  
--> <TOP OF FILE>  
...  
<BOTTOM OF FILE>
- 3) To enter a line of just a period:  
INPUT.  
u> ..  
u> .  
EDIT.  
u> C /.//  
--> .  
  
-or-  
  
INPUT.  
u> ^  
u> .  
EDIT.  
u> C /^//  
--> .

\* (change prompt toggle)

(change prompt toggle) \*

\*

Turn NETED's E: and I: prompting on or off.

The WHAT command shows the setting of this toggle. (Of course, if you have prompting, it is set; if you don't, it is reset.)

-----

Default: NETED,... -- set (E:, I: prompts)  
NETED,...,<any type> -- reset (no E:, I: prompts)

Toggles affecting the output of this statement: none.

Pointer after command completion: unchanged.

Related or similar commands: NETED second parameter.

Messages: none.

Examples:

```
u> NETED,NEWFILE
...
INPUT.
I:
u>   line 1
I:
u>   line 2
I:
u>   .
EDIT.
E:
u>   PA
   line 1
   line 2
E:
u>   *
u> c /2/2A/
   line 2A
u> ...
```

## \*\*\*\*\* A Summary of the NETED Commands by Function \*\*\*\*\*

The following terms are used in the description of NETED commands:

- <global> The letter "G" to indicate that the change is to be done to each occurrence in a line.
- <lfn> A local file. For commands which write a file, <lfn> cannot be an attached permanent file.
- <ncols> The number of columns affected by the command.
- <nlines> The number of lines to be processed by the command. In most commands, <nlines> may be a dollar sign ("\$"), which represents a very large number (9,999,999). In most commands, if <nlines> is preceded by a minus sign ("-"), the pointer is returned, after processing, to its position before the command was executed.
- <ntimes> The number of times the command is to be executed.
- <string> Any string of characters. Lower case letters will be converted to upper case (for non-ASCII types). Trailing blanks are not significant.
- <string1> (Change command only) Any string of characters. Lower  
<string2> case letters will be converted to upper case (for non-ASCII types). Trailing blanks are significant.
- <?> A question mark ("?") to allow you to verify each change before it is made.

The general format of a NETED command is:

[ <?> ] <cmd> [ <parameters> ]

## \*\*\* Toggle Switches \*\*\*

Page	Command and Description	
2-68	TRUNC	Automatic line truncation toggle. (Default: set (if <fmt> or <def_trunc_len> specified in NETED statement) reset (if both are omitted)
2-71	V	Verify toggle. (Default: set)
2-79	. (period)	Change-mode toggle. (Input-to-Edit or Edit-to-Input)
2-81	* (asterisk)	Turn NETED prompting (E: and I:) on and off.



\*\*\* Commands Which Move the Pointer \*\*\*  
\*\*\* Without Changing the Edit File \*\*\*

Page		Command and Description
2-5	B	Go to the bottom of the file.
2-16	F <string> FB <string>	Find the next (previous) line containing <string> starting in column 1.
2-30	L <string> LB <string>	Locate the next (previous) line containing <string> starting anywhere in the line.
2-42	N <nlines> N -<nlines>	Move the pointer <nlines> forward. Move the pointer <nlines> backward.
2-43	P <nlines>	Print lines single-spaced.
2-43	PD <nlines>	Print lines double-spaced.
2-64	T	Go to the top of the file.
2-42	<CR> (RETURN)	Go to the next line (same as "N 1").
2-42	- (minus)	Go to the previous line (same as "N -1").

## \*\*\* Print Commands Which Never Move the Pointer \*\*\*

Page	Command and Description	
2-18	FA <string> FBA <string>	Find all following or preceding lines which contain <string> starting in column 1.
2-31	LA <string> LBA <string>	Locate all following or preceding lines which contain <string> starting anywhere in the line.
2-39	LPL <page_length>	List page lengths of document in edit file.
2-45	PA	Print all lines single-spaced.
2-45	PDA	Print all lines double-spaced.
2-50	RA <nlines>	Print a range of lines around the current line.

## \*\*\* Commands Which Alter Lines \*\*\*

Page	Command and Description	
2-2	A <string>	Append <string> to the line. (Synonym: AR)
2-4	AL <string>	Add <string> to the start (left) of the line.
2-2	AR <string>	Add <string> to the end (right) of the line. (Synonym: A)
2-6	<?>C /<string1>/<string2>/ <nlines> <global>	Change <string1> to <string2>.
2-9	CENTER <nlines>	Center lines within LM-RM margins.
2-11	D <nlines>	Delete lines.
2-12	DTOP	Delete from the top of the file to the pointer.
2-13	DUP <ntimes>	Duplicate the current line <ntimes> times.
2-21	I <string>	Insert a line.
2-24	J	Join two lines putting one space between them.
2-24	JN	Join two lines putting no space between them.
2-26	K <ncols> <nlines> KL <ncols> <nlines>	Keep left-most columns.
2-28	KR <ncols> <nlines>	Keep right-most columns.
2-33	LEFT <nlines>	Left-justify lines within LM-RM margins.
2-49	R <string>	Replace a line.
2-51	<?>RC <nlines> <global>	Repeat the last change command.
2-53	RIGHT <nlines>	Right-justify lines within LM-RM margins.
2-57	SL <ncols> <nlines>	Shift left. (Synonym: TL)
2-58	SR <ncols> <nlines>	Shift right.

Page	Command and Description
------	-------------------------

2-66	TL <ncols> <nlines>
------	---------------------

Truncate on the left.  
(Synonym: SL)

2-67	TR <ncols> <nlines>
------	---------------------

Truncate on the right.

2-69	<?>UC <nlines> <global>
------	-------------------------

"Undo" the last change.

## \*\*\* File Manipulation Commands \*\*\*

Page	Command and Description	
2-40	M <lfm>	Merge a file into the edit file.
2-46	PF	All future writes will be to a permanent file device.
2-47	Q	All future writes will be to a queue device.
2-56	SAVE <lfm>	Save the edit file and terminate NETED.
2-73	W <lfm>	Save the edit file.
2-75	WL <nlines> <lfm>	Save a portion of the edit file.
2-77	WTOP <lfm>	Save from the top of the file to the pointer.

## \*\*\* Commands Which Terminate NETED \*\*\*

Page	Command and Description	
1-8	%A	Abort NETED. The edit file is lost!
2-48	QUIT	Terminate NETED without saving the edit file.
2-56	SAVE <lfm>	Save the edit file and then terminate NETED.

## \*\*\* Text/Word Processing Related Commands \*\*\*

Page	Command and Description	
2-9	CENTER <nlines>	Center lines within LM-RM margins.
2-23	IN <ncols>	Set the indent/undent from the left margin.
2-33	LEFT <nlines>	Left-justify lines within LM-RM margins.
2-35	LEN <nlines> <long>	List line lengths, optionally flagging long lines.
2-37	LM <lm>	Set the left largin for CENTER, LEFT, and RIGHT.
2-39	LPL <page_length>	List page lengths of the document being edited.
2-53	RIGHT <nlines>	Right-justify lines within LM-RM margins.
2-55	RM <lm>	Set the right largin for CENTER, LEFT, and RIGHT.

## \*\*\* Miscellaneous Commands \*\*\*

Page	Command and Description	
2-15	<?>ESC <e>	Change/display the "escape character".
2-20	H or HELP	Display a list of the current NETED commands.
2-36	<?>LFN <lfm>	Change/display the name of the edit file.
2-38	LN	Display the line number of the current line.
2-59	STAB <tabch> <tab1>,<tab2>,...,<tabn> STAB <tabch> <tab1>,<tab2>,...,<tabn>+<inc> STAB <tabch> <?>STAB	Set the tab character and tab stops.
2-62	STR STR <cmd>	Display the current string definitions or a specific definition.
2-65	TI	Show the elapsed, processor and I/O times and the processor time remaining.
2-74	WHAT	Display the edit file and NETED attributes in effect.
2-78	X <ncols>	Display a line of column markers.



## Appendix A

## \*\*\* NETED Messages \*\*\*

Listed in this Appendix are all the NETED messages, their causes and fix (if appropriate), and the command or commands which issue them.

Message	Cause and Fix	Command
<n> BAD CCS.	A count of the lines having other than "1", " ", "0", "-", "+" in column 1. To fix (assuming you are editing a document, shift the lines right.	LPL
BAD TAB LIST.	The tab list must be numeric and in ascending order. The first tab must not be 1.	STAB
<BOTTOM OF FILE>	The pointer is after the last line in a non-ASCII file.	
<Bottom of File>	The pointer is after the last line in an ASCII file.	
CANNOT DUP TOP- OR BOTTOM-OF-FILE.	You are using the DUP command while the pointer is just outside of the file. Move to the line you want to duplicate.	DUP
CANNOT JOIN EOR/EOF.	Either the first or second (or both) line to be joined is an end-of-record or an end-of-file.	J, JN
CANNOT JOIN LAST LINE.	The last line of the file cannot be joined because there is no next line.	J, JN

Message	Cause and Fix	Command
<hr/>		
<lf> - CANNOT OVERWRITE PERM FILE. The file you are editing is an attached permanent file and cannot be overwritten. Write or SAVE to another <lf>.		all writes
<lf> - CANNOT READ CONNECTED FILE. <lf> is connected to your terminal and cannot be read (other than for commands) by NETED. Edit or merge another file.		NETED, M
CURRENTLY DEFINED STRINGS: A/AR <len> ESC 3 <A/AR string> AL <len> ESC 4 <AL string> C 1 <len> ESC 1 <C string1> C 2 <len> ESC 2 <C string2> F/L <len> ESC 0 <F/L string> I <len> ESC 5 <I string> J/JN <len> ESC 7 <R string> P <len> ESC 8 <P string> R <len> ESC 6 <R string>  This is the output of the STR command. "STR <cmd>" will display only the string or strings for the requested <cmd>.		STR
DEFAULT TRUNCATION LENGTH=dt1 The value to be used to reset the right margin.		NETED, WHAT
DELETED TO BOTTOM. The number of lines you asked to delete was more than the number of lines left in the edit file.		D
E: The Edit Mode prompt. To turn off, use the "*" toggle. To resume the prompt, use the "*" toggle again.		Edit Mode

Message	Cause and Fix	Command
EDIT.	Indicates that you are now in Edit Mode.	NETED, .
EDIT FILE EMPTY.	There is nothing in the edit file to write!	all writes
<lf> EMPTY.	The file you are trying to merge is empty. Check the spelling of <lf>.	M
ENTER FILE NAME.	You did not specify a file name in the NETED command. Please do so now.	NETED
<n> EOR/EOFS.	A count of the lines containing only EOR or EOF and will become end-of-record and end-of-file marks when written to disk.	LPL
ERROR IN LN.	This error should never occur. If it does, please call User Services. It would be helpful if you could tell us what you were doing when you got this error.	LN
ERROR IN STAB.	This error should never occur. If it does, please call User Services. It would be helpful if you could tell us what you were doing when you got this error.	STAB
ESCAPE CH=<e>	<e> is the escape character currently in effect.	ESC, WHAT
ET= hh:mm:ss CP= nnn.nnn IO= nnn.nnn	Display, if verify is set, of the elapsed (wall clock) time, and processor and I/O times in seconds used during this NETED session.	QUIT, SAVE

Message	Cause and Fix	Command
ET= hh:mm:ss CP= nnn.nnn IO= nnn.nnn REMAINING CP= nnn.nnn Display of the elapsed (wall clock) time, and processor and I/O times in seconds used so far during this NETED session; and the processor time remaining. Be sure you have enough time left to save your edit file!		TI
FUTURE WRITES TO *PF. The SAVE, W, WL, AND WTOP commands will write onto a permanent file device so the file can be cataloged.		PF
FUTURE WRITES TO *Q. The SAVE, W, WL, AND WTOP commands will write onto a queue device so the file can be ROUTED to the input or output queue.		Q
***GET HELP--COMMAND ABORTED. You probably don't understand how to use the "?" in the change commands. Please see page 2-7.		?C, ?RC, ?UC
I> The Input Mode prompt. To turn off, use the "*" toggle. To resume the prompt, use the "*" toggle again.		Input Mode
ILLEGAL FILE NAME. The <lf> you specified may be too long, or start with a non-letter, or have certain special characters in it. Check your spelling.		NETED, M, all writes
ILLEGAL FILE TYPE. The <lf> you specified is neither a disk file nor a text file.		NETED, M, all writes
IMPROPER CHANGE ARGUMENT. Something is wrong with one of the fields of the Change command. Did you forget the closing delimiter or are there wrong characters following it? Re-enter the C command.		C

Message	Cause and Fix	Command
IMPROPER NETED ARGUMENT.	This is a valid command, but there is something wrong with one or more arguments. Re-enter the command correctly.	
INPUT.	Indicates that you are now in Input Mode.	NETED, .
INVALID DEFAULT TRUNCATION LENGTH - SET TO MAXIMUM.	The third execute parameter of the NETED statement (<def_trunc_len>) is zero or invalid and has been replaced by the maximum line length.	NETED
INVALID ESC ARGUMENT.	The ESC command accepts only a single character. Re-enter the command correctly.	ESC
INVALID NUMBER.	At least one number parameter has an invalid character (only digits, and perhaps a leading minus sign ("-")) are allowed.	D, DUP, IN, K, KL, KR, LEN, LM, RA, RM, SL, SR, TL, TR, X
--INVALID RESPONSE. MUST BE ONE OF: Y,C,N,Q.	You entered a response other than Y (yes), C (continue), N (no), or Q (quit). Enter one of these four letters.	?C, ?RC, ?UC
INVALID STR REQUEST.	STR recognizes only the following string identifiers: A, A/AR, AL, AR, C, C1, C2, F, I, L, P, and R, in either upper or lower case. Use a known value or omit it to display all strings.	STR
LFN: <lfm>	A display of the current name of the edit file.	LFN, WHAT

Message	Cause and Fix	Command
LFN: <lfm> <type> DEFAULT TRUNCATION LENGTH=140 <n> LINES TOGGLES SET: <toggles> LM=lll RM=rrr IN=+iii ESCAPE CH=<e> TAB CH=<tabch>	The output of the WHAT command.	WHAT
LINE TOO LONG. The maximum line length NETED supports is 140 characters. Either you are trying to edit or merge a file with longer lines, or a change created a line which was too long.		C, I, R
<n> LINES NOT CHANGED. Displayed if Verify is reset, since you don't see the changed and unchanged lines.		CENTER, LEFT, RIGHT
MERGE REQUIRES A FILE NAME. The "M" command must have an <lfm> specified.		M
<lfm> MERGED. Displayed when the specified file has been successfully merged into the edit file.		M
--NETED 2.0 The NETED header line.		NETED
NO ESCAPE CHARACTER. Displayed in response to ?ESC or WHAT if there is no active escape character.		ESC, WHAT
NO LINES TO WRITE. You tried WL while at bottom-of-file.		WL
NO OLD A/AR STRING. "A" or "AR" without a <string> may be used only if you have previously used A/AR with a <string>.		A/AR

Message	Cause and Fix	Command
NO OLD AL STRING.	"AL" without a <string> may be used only if you have previously used AL with a <string>.	AL
NO TAB CHARACTER.	Displayed in response to ?STAB or WHAT if there is no active tab character.	STAB
NOT A NETED REQUEST.	You have entered a "command" which NETED does not recognize. Check the spelling.	
NO TEXT BEFORE CURRENT LINE.	The pointer is at top-of-file or on the first line.	DTOP, WTOP
NOT FOUND.	The string you are searching for does not appear. Try the other direction. Check the search string you entered.	F, FA, FB, FBA, L, LA, LB, LBA
PG LEN * ><n>	The heading for the List Page Lengths command. Lines greater than <n> characters will be flagged.	LPL
SEQOPEN--710B	This error should never occur. If it does, please call User Services. It would be helpful if you could tell us what you were doing when you got this error.	NETED, M, all writes
SYNTAX ERROR.	Something is wrong. Check the syntax.	SL, SR, TL, TR
TAB CH=<tabch>	A display of the current tab character.	?STAB, WHAT

Message	Cause and Fix	Command
<n> TABS.	Too many tabs for the line length. This is the actual number of tabs defined. If this message is not displayed, then 30 tabs (or the special case of 1 tab) are defined.	STAB
<lfm> TOO LARGE FOR NETED.	The file you are trying to edit is too large for NETED to handle. Break the file into smaller pieces and edit each piece separately. See COPYRE, COPYS, and COPYSEL programs to split the file.	NETED, M
<TOP OF FILE>	The pointer is before the first line in a non-ASCII file.	
<Top of File>	The pointer is before the first line in an ASCII file.	
TRUNCATED.	A line longer than the truncation length (TRUNC set); or a line longer than the maximum line length (TRUNC reset) was truncated.	A/AR, AL, C, DUP, I, J, JN, K/KL, KR, R, RC, SL, SR, TL, TR, UC
TRUNC RESET. TRUNC SET.	One of these is displayed when NETED is initiated.	NETED
WL REQUIRES A FILENAME.	The WL command requires both the number of lines to write and the name of the file to which they are to be written. Re-enter the command with both arguments.	WL
<lfm> WRITTEN.	File <lfm> was successfully written and rewound.	all writes



Message	Cause and Fix	Command
Y--ACCEPT LINE		?C, ?RC,
C--ACCEPT LINE, CONTINUE COMMAND WITHOUT ?		?UC
N--DO NOT ACCEPT LINE, CONTINUE WITH COMMAND		
Q--QUIT THE COMMAND		
	If you give four invalid responses to these commands, this reminds you what may be entered. Enter a valid response.	
?>	The prompt for you to enter Y, C, N, or Q.	?C, ?RC, ?UC

## Appendix B

## \*\*\* Glossary \*\*\*

## Default truncation length

The value (stated or implied) from the NETED execute statement to be used for the initial truncation length / right margin and for resetting the right margin. See page 1-9.

## Edit file

The file you are editing. It may be an attached permanent file (which cannot be overwritten), an existing local file, or a non-existent local file (which will be created, at your option on a permanent file device or a queue device, when it is written).

## Escape character

Certain string commands and the print and join commands remember the strings. They can be referred to in other string commands by using the escape character and a digit. See page 1-7.

## Escape sequence

The escape character followed by a single digit used as a complete string in a string command. It refers to a string previously saved by another string command or by the print or join commands. See page 1-7.

## Left margin

Used for centering and left-justifying lines.

## Maximum line length

The length of the longest line supported by NETED. Currently, this is 140 characters.

## Right margin

Used for centering and right-justifying lines. It is also the truncation length when the TRUNC toggle is set. See page 1-9.

## Truncation length

Another name for the right margin. When the TRUNC toggle is set, changed lines which are longer than this length are truncated. Except for J and JN, the truncated information is probably lost. It may be hidden in one of the strings which were saved. Use the STR command to display them. See page 1-9.

## Appendix C

## \*\*\* Some Hints on Using NETED \*\*\*

1. If you enter NETED and misspell <lf>, you must be in Edit Mode to continue. If you are in Input Mode, enter a period (".") to go into Edit Mode. Then, either

- a) . enter "QUIT" to leave NETED.
  - . re-execute NETED with <lf> spelled correctly.
- b) .enter "LFN <lf>" to change the name of the edit file.
  - . if you are editing an existing file,
    - . enter "M <lf>" to merge the file.
    - . enter "T" to go to the top of the file.

## Appendix D

## \*\*\* New Features in NETED Version 2.0 \*\*\*

The following features of NETED are new at version 2.0:

1. There is a new execute statement and only one variety of NETED. NETEDF and NETEDR are now "NETED2,<lfm>,F" and "NETED2,<lfm>,R", respectively. Version 1 of NETED, NETEDF and NETEDR remain available. Throughout this manual, "NETED" refers to version 2, unless otherwise qualified.
2. The E> and I> prompts have been changed to E: and I: to remind you that this is the new version of NETED.
3. Lines may now be truncated automatically under control of the TRUNC toggle and the right margin (the NETED statement sets the initial values).
4. An "escape character" is available to facilitate the entry of previously defined strings in several commands. This character can be changed by the new ESC command. The STR command has been modified to show the associated number to use for substitution. Strings defined in the string commands (A/AR, AL, C, F/L, I, and R) are remembered and may be used in any of these commands. The last line printed by P or PD and the second line of a successful join (J/JN) are also remembered and may be used in the string commands.
5. Additional enhanced commands:
  - LEN - you can now flag long lines.
  - LPL - you can restrict the output including a display of only the totals.
  - STAB - has more pre-defined tab settings; can restore tabs to the default for the <type> parameter used in the execute statement; can display the tab character; can specify one or tabs with an increment; an invalid tab list will not change the current tabs.
  - TI - now displays the CP time remaining.
  - TL/TR - you can now truncate more than one line at a time.
  - WHAT - displays more information.
6. Additional new commands:
  - J, JN - join two lines with one or no space between.
  - K/KL, KR - keep the left- or right-most columns.
  - LFN - display or change the name of the edit file.
  - PF - all future writes will be to a permanent file device.
  - Q - all future writes will be to a queue device.
  - SL/SR - shift left or right.

Initial Distribution

Copies:

12 Director  
Defense Technical Information Center (DTIC)  
Cameron Station  
Alexandria, Virginia 23314

Center Distribution

Copies:

1	18/1809	Shoman, Dr. C. M.
1	1805	Cuthill, E. H.
2	1809S	
1	182	Camara, A. W.
1	184	Schot, J. W.
1	185	Schaffran, R.
1	187	Zubkoff, M. J.
1	189	Gray, G. R.
1	189.2	
1	189.3	Morris, J.
1	1893	Minor, L. R.
20	1893.1	Strickland, J. D.
10	1893.1	Sommer, D. V.
1	1895	Glover, A.
1	522	TIC (C)
1	522.2	TIC (A)
1	93	Patent Counsel

### DTNSRDC ISSUES THREE TYPES OF REPORTS:

1. **DTNSRDC reports, a formal series**, contain information of permanent technical value. They carry a consecutive numerical identification regardless of their classification or the originating department.
2. **Departmental reports, a semiformal series**, contain information of a preliminary, temporary, or proprietary nature or of limited interest or significance. They carry a departmental alphanumeric identification.
3. **Technical memoranda, an informal series**, contain technical documentation of limited use and interest. They are primarily working papers intended for internal use. They carry an identifying number which indicates their type and the numerical code of the originating department. Any distribution outside DTNSRDC must be approved by the head of the originating department on a case-by-case basis.

END

DATE

FILMED

8-88

DTIC